# Modeling of Laboratory Air Distribution Systems

**NETSAL & Associates**
**16182 Mount Lowe Circle**
**Fountain Valley, CA 92708**

**Principal Investigator:**

**September 4, 1999**

Robert J. Tsal, Ph.D.

# Table of Contents

## *1.0 Executive Summary*

Heating, ventilating and air conditioning (HVAC) systems are the major energy consumers in most industrial and commercial buildings. The HVAC system can cost up to 30% of the first cost of a building and 40% of the building's operating energy expense. Proper system design is one of the ways of reducing energy use and life cycle costs of a facility, as well as enhancing the performance of HVAC systems.

The object of this project was to develop a concept for a computerized design and analysis program for multi-fan air distribution systems in research laboratories. A numerical model based on this concept will be capable of analyzing HVAC equipment, control elements, laboratory equipment, and laboratory spaces as one generalized network system. Constant air volume, variable air volume, and central supply air with multiple exhausts are systems that were considered. The project also considered all three major parts of an airflow system; the supply fan and ducts, the laboratory spaces, and the exhaust fan(s) and ducts; as a general system with loops. Additionally, the long-term, near-term, and short-term goals, potential users, and a plan for software development, testing and implementation, as well as the commercialization strategy were also considered.

The project was devoted to two major duct design goals: optimization and simulation. The optimization goal is to select duct cross-sections, fans, and duct mounted equipment that minimizes the life cycle cost that includes both energy and capital cost. The simulation (modeling) goal is to determine the air flows and pressures at steady-state condition after initial data such as the position of fume hood sashes have been fixed and the control sensors, controllers, and actuators have already established stability. The necessity for simulation appears in many HVAC designs including determining operating performance, investigating system stability under different operating conditions, and pressure and airflow balancing after a system modification. A statistical analysis of opening fume hood sashes based on their operating schedules will allow an analysis of the effect of pressurization on the laboratories and an evaluation of the efficiency of the generalized air distribution system.

The network optimization problem requires finding a minimization solution for nonlinear objective function and nonlinear and integer constraints. There are several nonlinear mathematical programming methods for solving such problems. The network simulation problem requires a solution for a large system of simultaneous nonlinear algebraic equations. Numerous methods capable of solving such equations have been studied. The most promising approaches are the family of Newton Affine Invariant methods, the Tensor methods, and the Homotopy methods. All three approaches are recommended to be included into the computer code and selected automatically during the calculation. An evaluation was conducted for two software packages developed by Los Alamos National Laboratory (LANL), one for solving a system of simultaneous nonlinear equations (Spark) and the other for nonlinear optimization (GenOpt).

## *2.0 Abstract*

The object of this project was to develop a concept for computer optimization and modeling of multi-fan air distribution systems in research laboratories. A numerical model based on this concept will be capable of analyzing HVAC equipment, control elements, laboratory equipment, and laboratory spaces as one generalized network system. Constant air volume, variable air volume, central supply air with many exhausts are systems that will be considered. The optimization goal is to select duct cross-sections, fans, and duct mounted equipment that minimizes the life cycle cost that includes both energy and capital cost. The modeling goal is to determine the air flows and pressures in a steady-state condition after initial data

such as the position of fume hood sashes have been fixed and the control sensors, controllers, and actuators have already established stability. Statistical analyses of opening fume hood sashes based on their operational schedules will allow the determination of the effect of pressurization on the laboratories and the evaluation of the efficiency of the generalized air distribution system.

The network optimization challenge is to find a minimization solution for the nonlinear objective function and nonlinear and integer constraints. There are several nonlinear mathematical programming methods for solving such problems. The network simulation problem requires a solution for a large system of simultaneous nonlinear algebraic equations. Numerous methods capable of solving such equations were studied. The most promising approaches are the family of Newton Affine Invariant methods, the Tensor methods, and the Homotopy methods. If a good initial guess is known, the best choice is one of the Newton Affine Invariant methods. For a comparatively large problem, when time for solution is significant, the Tensor methods are a good choice. For most practical cases, when either: (1) time is deficient to search for a good initial guess, (2) automation is needed for different types of technical solution evaluations, or (3) parametric evaluation is necessary, Homotopy methods are the best. All three approaches are expected to be included as a part of the computer code and selected automatically during the calculation.

This project consists of the implementation plan for developing a multi-fan laboratory duct design computer program, including computer code, data base population, testing, and support. Also, an evaluation was conducted for two software packages developed by LANL one for solving a system of simultaneous nonlinear equations (Spark) and the other for nonlinear optimization (GenOpt). This study shows that for the stated goals the use of neither of these packages is efficient. This report identifies long-term, near-term, and short-term goals; potential users; and includes software development, testing and implementation plans, as well as the pricing strategy.

## *3.0 Introduction*

Studies show that HVAC systems are a major energy consumer in laboratory buildings. The HVAC system can cost up to 30% of the first cost of a building and 40% of the building operating energy expense. Optimum HVAC design reduces life cycle cost and enhances HVAC system performance. Actual operating conditions in a laboratory vary and can result in an airflow imbalance and destabilized HVAC operation. Such operating conditions can negatively impact work being performed in a laboratory. Computerized numerical modeling can predict actual operating conditions and help prevent negative impacts.

This project developed the concepts and implementation plan for optimization and modeling of airflow systems, including the principles of duct design software development as long-term, near-term, and short-term goals.

The long-term goal is to develop a computerized numerical model capable of optimizing large multi-fan air conditioning systems that include many individual exhaust systems. In addition to optimization, this computer program will simulate HVAC airflow distribution systems and analyze actual conditions allowing "what-if" scenarios to be performed. This capability is ideally suited to evaluate airflow systems for laboratories with changing missions.

Three major calculation steps are considered: (1) Optimization, (2) Testing and Balancing (TAB), and (3) Simulation.

The design program for airflow distribution systems will perform both economic optimization and comprehensive airflow distribution analyses for laboratories, including the following elements:

(1) Optimization by selecting fans, duct mounted equipment, and ductwork sizes that minimizes the system life cycle cost,
(2) Selecting and identifying the required adjustment to duct balancing dampers to obtain design air flows,
(3) Simulation of duct systems in order to evaluate air flows, velocities, pressures, and temperatures in every duct section and each room under various process, weather, and occupancy conditions,
(4) Performing statistical analyses for positioning fume hood sashes in order to obtain a usage factor,
(5) Combining optimization and simulation procedures with acoustical design and analysis capabilities,
(6) Combining simulation systems with the effects of airflow migration between rooms due to internal and external conditions such as wind, buoyancy, internal and external pressures, zoning, etc.

The near-term goal is to develop and implement a multi-fan simulation program capable of optimizing and modeling single supply, many laboratories, and many exhaust systems as one generalized system. A candidate tools for solving the nonlinear mathematical programming problem and nonlinear algebraic equations necessary to optimize and simulate ductwork systems was LBNL's GenOpt and Spark software. The near-term goal was to evaluate how GenOpt and Spark can be utilized as the main software packages to solve these problems.

The short-term goal is to develop an optimization computer program as a first part of the duct design software. It will either utilize the T-Method, as presented by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), or a more advanced new method, which considers the laboratories as a part of a duct network.

Two variations of the T-Method are capable of solving both simulation and optimization duct design problems. The simulation part of the T-Method has been implemented by the T-Duct computer code (T-DUCT 1994) for ductwork modeling. However the T-Method is capable of optimizing only a single fan duct system. All existing commercial duct design computer codes usually on arbitrary engineering procedures such as Static Regain or Equal Friction and do not accomplish life cycle cost optimization.

## *4.0 Background*

Proper airflow distribution system design is one of the most efficient ways of reducing life cycle cost and enhancing the performance of air conditioning systems. Actual operating conditions in a laboratory served by multiple supply and exhaust systems vary substantially and can result in under-pressurization or over-pressurization and air flow destabilization beyond design limits. Such conditions are inefficient and can be unsafe and negatively impact the work being done in the laboratories. Engineers have no practical tools to calculate actual airflows, pressures, fan operating points, and the effect of system control in multi-fan laboratory air distribution systems.

Negative pressure confinements are very important for many laboratories dealing with harmful substances, such as nuclear materials or toxic chemicals. Positive pressure confinements are used in "clean rooms" in order to prevent them from becoming contaminated from an external source. Negative or positive laboratory pressure creates the driving force for either air infiltration or exfiltration through doorway slits, construction cracks, etc. On the other hand, a high pressure differential between workspaces is also undesirable. If infiltration or exfiltration exceeds the design limit, it may result in loss of proper room and/or zone pressurization, substantial heat and energy loss with considerable dust and dirt carryover, and airflow destabilization. The need to calculate airflow distribution, internal static pressure, and control system response occurs early in the design phase or any time an engineer is studying the effect of airflow system performance. The necessity for simulation appears in many air system designs and analyses, such as determining system operating performance, investigating system stability under varying operating conditions, system retrofitting, emergency conditions, accidents, fire/smoke protection systems, pressure and airflow balancing after system modification, and equipment failure.

Multiple laboratories located in the same building are usually served by one or more common supply systems and multiple exhaust systems that can be individual or manifolded into a centralized system. Topologically the laboratory air distribution systems can be represented by a number of isomorphic tree-graphs connected at the terminals.

### *5.0 Objectives*

The major long term goal of this project is the development and commercialization of a software package for multi-fan and duct air distribution system for laboratories, called BELLAIR. This package includes two major parts: optimization and modeling (simulation). The object of this project was to develop a concept and implementation plan for a computer code for the optimization and modeling of air distribution systems in research laboratories. A numerical model based on this concept will be capable of: (1) selecting fans and duct cross-sections, and (2) analyzing all aspects of the air distribution, including HVAC equipment, ducts, control elements, laboratory equipment, and laboratory spaces, as one generalized system. Constant air volume, variable air volume, central supply air with multiple exhausts or manifold exhaust systems were considered.

The optimization part of proposed computer program, BELLAIR, will allow HVAC engineers to select fans and duct cross-sections.

The modeling part of the program BELLAIR will calculate actual airflows, pressures, and fan operating points in laboratory multi-fan air distribution systems. It will also allow HVAC engineers to predict actual static pressures in laboratory spaces at different operating conditions where space pressurization, confinement zoning, and flow/pressure stability are the most important requirements. The modeling goal is to determine air flows, velocities, and pressures at steady-state conditions after the initial data such as the position of fume hood sashes have been established, and the control sensors, controllers, and actuators have been stabilized. This includes determination of the various damper positions and the laboratory space pressures relative to a reference pressure. The code will provide statistical analyses of fume hood sash positions based on their operating schedules and identify the conditions when laboratory spaces are under-pressurized, over-pressurized or destabilized. The results of such analyses can be used to evaluate the efficiency of air distribution systems and to calculate the fume hood/biological safety cabinet (BSC) usage factors.

BELLAIR will be capable of modeling the control of each fan or damper from a remote sensor located anywhere in the system selected by the user including ducts, laboratories, cabinets, exhaust stack discharge, etc. The control parameters could be static pressure, static pressure difference, airflow volume, air velocity, temperature, or temperature difference.

This will be the first development of a practical engineering tool for multi-fan multi-laboratory air distribution system simulation. This tool will combine airflow distribution with control system and statistical analysis of equipment loading.

Quality multi-fan system design can be achieved only by using a comprehensive computer program. This project was targeted to the development and implementation of such a computer program. The proposed program would be used by HVAC engineers in the design and retrofitting of air distribution systems in research laboratories. The simulation part of the program would allow an HVAC engineer to calculate actual airflows, pressures, and fan operating points in laboratory multi-fan systems as well as the static pressure in laboratory spaces at different operating conditions where space pressurization, confinement pressure zoning, and flow/pressure stability are the most important requirements. Such a computer simulation program would allow the engineer to play "what-if" scenarios that are most important in providing maximum flexibility for laboratories with changing technology and requirements.

The plan is to distribute this computer program through the Internet where it will serve companies and individuals involved in the design, testing, and balancing of air conditioning systems for laboratories. The laboratory staff responsible for HVAC systems maintenance will also be served.

There are four major steps of multi-fan system simulation:

(1) defining airflows and pressures for initially setting control devices,
(2) adjustment of positions of control devices in accordance with their setting,
(3) random selection of the positions of fume hood sashes based on their loading schedules,
(4) system performance analysis.

As a result of such calculation, the user receives airflows and velocities in all system sections and pressures at all system nodes including workspaces, manifolds, fume hood inlet sashes, stack exhausts, and ductwork. The user also receives fan operating points, positions of all system control devices, and electrical energy consumption. Air system performance can be studied by randomly selected fume hood sash positions and other laboratory unit variables. As soon as the user inputs the loading schedule for each hood, the computer will randomize the position of sashes and calculate the total usage (diversity) factor as a recommendation for optimum design.

The four major fragments of the BELLAIR program are the:

(1) preprocessor that contains data input, verification, and printout,
(2) solver that includes calculation routines,
(3) postprocessor that performs verification and printout of the results,
(4) populated database.

The main platform for the BELLAIR code will be MS Windows 95/98/NT for IBM PC computers. The programming language will be C++ with the recommended compiler, Visual C++. A demo of the program is planned to be installed on the Internet. The program must be extremely user-friendly, since it will be used by HVAC specialists who are knowledgeable in duct design but are not knowledgeable in duct optimization and simulation. Data input should be checked when it is being entered. The error messages should be divided into two classes, fatal and warning. They should describe every error and display recommendations for correction. Full capability of inserting, copying, moving, or deleting any part of the data should be implemented by using data blocking. Help screens should include examples. Graphic representation should be used for fitting selection. The practicality of graphic input for topology and length has to be investigated. The main steps of the development and implementation of the BELLAIR program are the architecture, coding and debugging, testing, data base development and population, program installation, Demo, user manual, and program support.

The implementation plan for developing a computer code for the optimization and modeling of air distribution systems in research laboratories describes the following two phases:

**Phase I** includes the development and/or selection of numerical methods for solving nonlinear mathematical programming problems and large systems of simultaneous nonlinear algebraic equations,

**Phase II** includes computer code writing, debugging, testing; database development and population; and program implementation and support.

The proposed project is divided into the following parts:

**Part 1.** Development and/or selection of the mathematical methods for solving large nonlinear mathematical programming problems and large systems of nonlinear algebraic equations including:

(1) Problem definition
(2) Analysis of existing methods
(3) Problem formalization
(4) Numerical model development and convergence study
(5) Software implementation plan development

Part 1 [selecting a method for solving simultaneous nonlinear algebraic equations was completed as part of this project in December 1998 under California Institute of Energy Efficiency (CIEE) Project #4910810, "Numerical Modeling of Multi-fan Air Distribution Systems for Research Laboratories." The study resulted in the selection of the following numerical methods for implementation: (1) Affine Invariant Newton methods, (2) Tensor methods, and (3) Homotopy methods.

A study of the existing computer program, SPARK, developed by LANL for solving a large system of nonlinear simultaneous algebraic equations was one of the subjects of this project. This study shows that using SPARK is unsuitable because it will increase the difficulties when solving large systems of simultaneous equations.

**Part 2. C**omputer code writing, debugging, and testing includes:

Program Architecture and Graphical User Interfase (GUI) Preprocessor
(1) Program architecture development
(2) Topology representation
(3) Screens and menus
(4) Data checking and diagnostic
(5) On-line error messages
(6) Data manipulation routines (browse/insert/copy/move/delete)
(7) Information accessed from the database
(8) Help indexes, screens, and messages
(9) Input data listing
(10) Classes

Solver

(1) Algorithm
(2) Local fitting library access
(3) Element data access
(4) Hydraulics
(5) Control elements response
(6) Heat loss or gain
(7) Air leakage
(8) Energy consumption
(9) Newton method
(10) Tensor method
(11) Homotopy method
(12) Statistical data selection and recording
(13) Classes
(14) Output data storing

Postprocessor

(1) Output screens
(2) Graphical fan-system characteristics
(3) Graphical pressure diagrams
(4) Printout of the results

**Part 3.** Data base development and population

(1) Local database development
(2) Development and population of central fittings database
(3) Element data tables
(4) Program messages (data-driven information)
(5) Graphical representation of fittings and elements

**Part 4**. Program implementation and support

(1) Program installation
(2) Development of installation routines using WISE

**Part 5**. DEMO development (for Internet)

(1) Architecture
(2) Algorithm
(3) Screens
(4) Code writing/debugging/testing
(5) Server downloading

**Part 6**. Writing the manual

(1) Program overview
(2) Hardware requirements
(3) Program installation
(4) Quick start

(5) Practicing with program
(6) Run and output
(7) Examples
(8) Data verification and validation

**Part 7**. Testing
(1) Alpha testing
(2) Beta testing
(3) Code correction and modification

**Part 8**. Support
(1) Performing software house functions
(2) Telephone support
(3) E-mail/fax written support
(4) Update the code and the database.

### *6.0 Potential Users*

The major areas of application for the BELLAIR computer program include consulting engineering in the HVAC field, TAB personnel, and maintenance engineering. The primary beneficiaries of the use of both optimization and simulation programs are the owner of the building and the lessee company or occupant.

There are a large number of **potential users** of this program, including:

- HVAC consulting engineers and designers,
- TAB specialist,
- research & development specialists in HVAC,
- fire protection engineers,
- equipment manufacturing engineers,
- utility company energy conservation engineers,
- HVAC plant engineers.

### *7.0 Flow Distribution Criteria and Calculation Methods*

Three major principles are the basis for flow distribution in a duct system: equilibrium of mass, equilibrium of energy, and equilibrium of momentum. These principles are the foundation for pressure and flow balancing in a duct system. Pressure balancing requires that the sum of total pressure losses in every branch of the system is equal to the total fan pressure. The relationship between air flows, pressure losses, and resistances for systems that includes cycles is similar to the first and second **Kirchoff** laws for electrical networks.

Any ductwork system will automatically balance itself. This means that the amount of air the system will move is such that the energy consumed by the fan to move this amount of air will be equal to the energy loss due to ductwork friction, dynamic loss, and change of air properties.

The Darcy-Weisbach equation for round and rectangular ducts represents the main hydraulic characteristic of a duct. The continuity equation is applied as a function of air velocity, airflow, and duct cross-section area.  Air density is a function of air temperature and pressure.  The friction coefficient is calculated by using Colebrook's equation as a function of Reynolds number, duct roughness,  and hydraulic diameter. The Reynolds number itself is a function of kinematic viscosity, diameter and velocity.  Because Colebrook's equation cannot be solved explicitly,  it was simplified by Altshul-Tsal (ASHRAE 1997).

Research shows that leakage in an assembled duct can be estimated by an exponential function of static pressure difference within a duct and surrounding air.

Typically, the duct internal static pressure is constantly changing from the static pressure at the fan to the terminal inlets or outlets due to friction, dynamic losses (fittings), and leakage.  Prior to the pressure loss calculation it is unknown what the static pressure will be at each duct section.  Fitting resistance is calculated as a part of the Darcy-Weisbach equation.  It can be represented by the sum of local resistance C-coefficients in a duct section. Sections can be divided between fittings to obtain a higher level accuracy of the duct leakage calculation.  Considering that duct leakage is a function of static pressure which depends on the location of fittings, Tsal, Behls, and Varvak (Tsal 1998) suggested defining any section as a part of a duct between two fittings, instead of using the sum of C-coefficients as in the Darcy-Weisbach equation.

There are a large number of tables and formulas that are used to calculate C-coefficients for fittings. The most valuable source of C-coefficients is contained in the ASHRAE Duct Fitting Database (DFDB 1994) and the Handbook of Hydraulic Resistance (Idelchik 1996).

The ventilation system in research laboratories can be served by a large number of supply and exhaust fans.  The BELLAIR computer program will be capable of optimizing and modeling a system with many fans.  Fan characteristics are a function of fan total pressure and fan discharge airflow for each rotational speed.

There are a number of control strategies for a fan:

(1) control by a damper,
(2) control by fan rotation,
(3) control by pitch,
(4) control by an inlet vane.

A network optimization problem is considered to be solved when all necessary fans and duct cross-sections have been selected and the life cycle cost obtained is minimum.

A network simulation problem is considered to be solved when the iteration process that describes steady-state hydraulic and control conditions has converged.  Such a problem requires a solution for a large system of simultaneous nonlinear algebraic equations.  Therefore, one of the main goals of this project was the selection of a numerical method for solving such equations.  Numerous methods capable of solving nonlinear algebraic equations for both branched and cycled network simulation have been studied (Tsal and Behls 1986). For branched systems these include: Equivalent Nozzles, Unit Flow, Duct Characteristic, Equivalent Resistance, Steepest Descent, and T-Method with air leakage.  For cycled systems the most promising approaches are the family of Newton Affine Invariant methods, the
Tensor methods, and the Homotopy methods. If a good initial guess is known, the best choice is one of the Newton Affine Invariant methods.  For a comparatively large problem, when time for solution is significant, the Tensor methods are a good choice. For most practical cases, when either time is deficient to search for a good initial guess, automation is needed for different types of technical solutions evaluation, or parametric evaluation is necessary, Homotopy methods are the best.  All three approaches (Newton

Affine Invariant, Tensor and Homotopy) are expected to be included into the computer code with the best approach being selected automatically during the calculation.

All methods for solving nonlinear equations involve the solution of many sets of linear algebraic equations. For small tasks it is possible to use simple methods based on direct solution techniques. Wherever there is a sparse matrix, sparse mode elimination techniques may be successfully used. For large and super large tasks Krylov's (see Huang and O'Leary 1994) method should be applied.

Control system simulation requires the use of a mathematical programming technique to get the minimization of an objective function that describes residuals. Vicente's (1996) algorithm is suggested for solving this problem.

### *8.0 Airflow Optimization Criteria*

There are two alternatives for a well designed duct system. The first is to size the ductwork and select a fan in such a way that air delivery meets required design airflow rates. This is called duct sizing. The second alternative is to select duct sizes and a fan by minimizing the life cycle cost that include the initial installation cost and the operating cost for the expected life for the system. This is called duct optimization.

The goal of any duct optimization method is to determine duct sizes and select a fan that minimizes system life cycle cost. The owning cost includes initial cost, property taxes, insurance, and salvage value. The operating cost includes the costs of energy, maintenance, operating labor, taxes, and cost escalation. The purpose of optimization is to compare system costs for different fan total pressures; therefore, many of the above factors are constant and can be excluded from the objective function. Only initial cost, energy cost, time period, escalation rate, and interest rate are needed for optimization.

Two situations are under consideration. Where the fan is unknown, the requirement is the selection of duct sizes, optimal system pressure loss, fan, and motor size. If the fan and motor are existing, and the fan characteristic is known, the requirement is to size ducts through the optimum distribution of pressure throughout the system. This minimizes ductwork construction cost for an existing fan and motor.

Electrical energy cost depends upon residential, commercial, and industrial retail prices of electricity as well as upon the differences in demand and consumption costs. The data in the Electric Power Annual are adjusted for a 40 kW demand for commercial consumers and a 500 kW demand for industrial consumers. It must be remembered that the electric demand cost must be paid, not only for HVAC systems, but for the entire building and that the difference between the fan pressures for optimized and non-optimized systems is a small part of this demand. Therefore, to simplify the calculation procedure it is assumed to be a constant.

The important economic data used in optimization are interest rate, amortization period, and duct cost. If the interest rate is unknown, the recommended interest rate will be 6%. If the amortization period is unknown, 10 years will be used. The initial cost includes the cost of ducts and the HVAC equipment. The duct cost is presented as a function of the cost per unit area of duct surface, adjusted for straight duct and fittings.

Heating and cooling loads depend on many probability factors. The accuracy of any calculation should not be greater than the accuracy of the input data. To some degree ducts are always installed differently than they are designed. The economic data are good only for the current period and cannot predict situations like oil embargoes or Persian Gulf crises. Therefore, there is no need for over-accurate economic data for duct design.

Installed duct prices per unit area of ductwork are available from Sheet Metal Estimating catalogs. A more accurate optimization can be obtained by separating the cost of straight ducts from fittings.

Equipment included in the objective function is primarily the central air-handling unit and any duct mounted components. The pressure loss of duct mounted equipment (coils, silencers, terminal control units) is recognized as an additional pressure loss in those duct sections where this equipment is located.

An important factor of duct optimization is the cost of space required by ducts and equipment. This additional space consideration may reduce the size of ducts and thereby increase energy consumption. If saved space is available for use as rentable area, it must be included into the objective function. Since this saved space usually cannot be utilized, the space cost can usually be ignored.

An examination of the electrical energy retail prices shows significant variations. The difference in electric energy cost between residential customers in New York City and in Seattle is a factor of 8.7 to 1. The extreme difference in energy cost over the USA for industrial consumers is a factor of 31.3 to 1 between Saint Paul city, Alaska state (50.66 c/kWh) and Douglas county, Washington state (1.62 c/kWh).

The extreme cost for ductwork is $3.10 per square foot for 26 gauge spiral ducts and $12.02 per square foot for 10 gauge galvanized duct, a ratio of 3.9 to 1. Combining the two ratios yields potentially higher ratios based upon locale and type of ductwork. Due to the electrical energy and ductwork price variations there is a great potential for reducing duct system cost through optimization.

Following is the list of constraints that must be satisfied for duct optimization. A detailed explanation of each constraint can be found in the paper by Tsal and Adler (1987).

- Kirchoff's first law. For each node the airflow in is equal to the airflow out.

- Pressure balancing. The total pressure loss in each path must be equal to the fan total pressure. A different interpretation of the same constraint is: for every node the total pressure losses for all paths are equal. This restriction sometimes is called Kirchoff's second law and is analogous to electric cycle networks.

- Nominal duct sizes. Each diameter of a round duct or height and width of a rectangular duct is rounded to the approximate lower or higher nominal size. Nominal duct size normally depends on the manufacturer's or contractor’s standard increments. Such increments depend on the country where the duct system has to be installed.

- Air velocity restriction. This is an acoustic or ductwork regenerated noise when too high or particle conveyance limitation when too low.

- Preselected sizes. Duct diameters, heights, and/or widths can be preselected.

- Construction restrictions. Architectural space limitations will generally restrict duct sizes.

- Equipment. Central air-handling units and duct-mounted equipment are selected from the set produced by industry.

## *9.0 Calculation Methods and Software Tools*

### *9.1 Duct design procedure*

The general procedure for HVAC system duct design is as follows:

(1) Study the building plans, and arrange the supply and return outlets to provide proper distribution of air within each space. Adjust calculated air quantities for duct heat gains or losses and duct leakage. Also, adjust the supply, return, and/or exhaust air quantities to meet space pressurization requirements.

(2) Sketch the duct system, connecting supply outlets and return inlets with the air-handing unit or air conditioner. Space allocated for supply and return ducts often dictates system layout, ductwork shape, and duct routing.

(3) Divide the system into sections and number each section. A duct system should be divided at each point where flow, size, or shape changes. Locate the section fittings toward the supply and return (or exhaust) terminals.

(4) Size ducts by the selected compute program. Calculate system total pressure loss and then select the fan. For optimization fan selection and duct sizing are a common procedure.

(5) Layout the system in detail. If duct routing and fittings vary significantly from the original design, recalculate the pressure losses. Reselect the fan if necessary.

(6) Resize duct sections to approximately balance pressures at each junction. For optimization this procedure is a part of the previous steps.

*(7)* Analyze the design for objectionable noise levels, and specify sound attenuators as necessary. The presented procedure is based on manual calculation or computerized duct sizing.

### *9.2 Branched systems*

#### *9.2.1 Simulation*

There are several numerical methods and computer programs for calculating flow distribution in a branched duct system. The oldest flow simulation method, called **equivalent nozzles**, was developed in Germany at the end of the 19th century by Bless (see Lobaev 1959, p.79). The intent of this method is to substitute the resistance of ductwork by the equivalent resistance of a nozzle. The cross-section of the nozzle is then calculated in such a way that for the same flow, the pressure losses in the system and at the nozzle are the same. Then, by computing the flow at this nozzle from the known pressure difference, the flow through each section can be calculated. The method is oriented to the quadratic law of resistance, which is seldom used for HVAC duct design.

The **unit flow** method was developed by Kamenev (1938). This method is used to calculate airflow distribution in an existing system and assumes that flow through the terminal section is equal to one unit of flow. Then the pressure loss per one flow unit is calculated for sections connected in parallel and in series. Flow is proportional to the number of units for each section. This method, like the **equivalent nozzle** method, is used in cases of the quadratic law of resistance.

The **duct characteristic** method was developed by Butakov (1949). Butakov used the old friction coefficient formula developed by Bless and substituted it into the Darcy-Weisbach equation. Then he substituted lengths, C-coefficients, and diameters for coefficients called duct characteristics. He then derived formulas for calculating these coefficients for sections connected in parallel and in series. This method also can be used only for ducts operating in the quadratic law friction range.

The **equivalent resistance** method was refined by Lobaev (1959) and can be used for duct sizing and system simulation. This method is almost the same as the **duct characteristic** method. Instead of using Bless's formula for calculating friction coefficients, Lobaev applied his simplified formula for metal ducts to obtain a function of hydraulic similarity that depends on friction coefficient, length, diameter, and C-coefficient.

The **steepest descent** method was applied for duct simulation by Tsal and Shor (1967) and implemented as a computer program. The minimized objective function is the sum of the squares difference between fan pressure and pressure losses in branches. The airflow rates are the unknown variables in a system of nonlinear equations. The gradient is defined as a result of calculating a matrix of partial derivatives. The descent step is normalized at each iteration as a function of the maximum gradient vector. If the gradient is positive, it is divided by two. The authors reported the development of a computer program for calculating the flow distribution in branches, correcting the fan operating point in the case of a change of flow, and calculating the required brake horsepower. Major applications include industrial exhaust systems conveying particulate matter where dampers are prohibited. On the basis of this method R.J.Tsal created and successfully used for many years a computer program, which simulates ventilation systems (Tsal 1967). This program not only allows one to solve the problem of airflow distribution, but also corrects fan pressure in accordance with the characteristic of the fan when the flow changes.

Tsal and Chechik (1968) developed the algorithm for flow distribution, based on Bellman's **dynamic programming** method. The authors noted that this method is more difficult for implementation than the steepest descent method, but it has no convergence problems. The duct system is divided into a number of stages. Different pressure levels are assigned for each stage, and the method is based on a variation of flows at each pressure level. Only flows that correspond to the same pressure losses at branches are stored. Fan pressure is analyzed when the dynamic calculation process reaches the root section.

The **T-Method**, based on Bellman's Dynamic Programming ideas, has been developed by the author in cooperation with ASHRAE as a design tool for a single-root duct system simulation (Tsal 1990). T-Method (ASHRAE 1997) is based on the same tee-staging idea as Dynamic Programming (Bellman 1957; Tsal 1967; Tsal 1968). The T-Method incorporates the following major procedures:

(1) System condensing. Condenses the branched tree system into a single imaginary duct section with identical hydraulic characteristics and the same owning cost as the entire system.
(2) Selection of an operating point. Determines system flow and pressure by locating intersection of fan and system curves.
(3) System expansion. Expands the condensed imaginary duct section into the original system with flow distribution.

The T-method duct simulation determines the flow within each section of a duct system for known duct sizes and fan characteristics. The shortcoming of the T-Method is that it calculates tree-networks only. A computer program, T-Duct, based on T-Method was developed by R.J.Tsal in 1994 and is being successfully used for simulating duct systems with a tree-graph topology and a single fan located at the root. Tsal (1998) has recently developed and presented the T-Method with Duct Leakage to ASHRAE. The purpose of T-method simulation is to determine the flow within each section of a branched duct system of known duct sizes and fan characteristics.

### *9.2.2 Optimization*

Many analytical and numerical methods for pipe and duct optimization have been developed during the last century. A comprehensive survey of the existing numerical duct optimization methods was conducted by Tsal and Adler (1987). The first optimization method was developed by Grashoff in 1875 for a single pipeline. Several calculation procedures attempt to minimize total cost by establishing optimum velocities or friction rates. These procedures are based on the classical calculus minimization technique of setting the first derivative to zero in order to find the diameter of the pipe or duct.

The classical method of optimization for a multipath district heating system was first applied by Shifrinson (1937) and for [multi-path] duct systems by Lobaev (1959). Even though these works were done before the computer era, their techniques are impractical for manual calculation. According to Tsal and Adler (1987), analytical approaches may be effectively used only to identify trends in system behavior. A comprehensive analysis of such a duct system was published by Bouwman (1982).

The computer-aided numerical optimization methods are divided into two categories, discrete methods (coordinate descent, dynamic programming, and T-Method) and continuous methods (penalty function, Lagrange multipliers, reduced gradient, and quadratic search).

The coordinate descent method is the most common technique implemented for duct optimization (Tsal 1968). Dynamic programming is one of the most powerful methods applied for multipath tree network optimization by Tsal and Chechik (1968). The penalty function method transforms a constrained problem into a non-constrained problem by adding penalty coefficients to the objective function. This method was applied for different networks by Tsal and Chechik (1968). Another way to directly optimize a network is by the Lagrange multipliers method (Zanfirov 1933, Bertschi 1969, Kovaric 1971, Stoecker 1971). The modified Lagrange multipliers method was applied for network optimization by Murtagh (1972). Reduced gradient (Arklin 1979) is one of the best computerized techniques for rectangular duct optimization. It performs nonlinear optimization with equality constraints and then applies the Newton-Raphson technique to find an optimum solution. A technique called quadratic search was introduced for a concave problem optimization by Leah (1987). It was applied for chilled water system optimization.

Many of these methods are capable of finding the minimum of an unconstrained concave problem, but most fail to yield a successful solution that can be used in practice. In general, the objective function is not uniformly concave. An example in the Tsal and Adler paper (1987) explains this phenomena.

An advanced optimization technique for duct design is being developed (Tsal 1986). The T-Method is based on the same tee-staging idea as dynamic programming (Bellman 1957, Tsal 1968). The important advantage of the T-Method is that it is capable of handling constrained optimization processes including non-linearity and integer duct size rounding.

The last enhancement of the T-Method shows that it is capable of optimizing a duct system with air leakage (Tsal 1998). The T-Method converges efficiently. Usually, five iterations are sufficient to obtain the optimum solution with a high degree of accuracy.

The shortcoming of the T-Method is its difficulties in optimizing cycle systems.

### *9.3 Cycle systems*

The first method for water distribution in cycle systems was developed by Andriyashev and published in

1932 (Andriyashev 1932). In two years Lobachev (1934) published different technique. Hardy Cross from the Illinois University published his method in 1936 (Cross 1936). The goal of all these methods is to find the air/water/gas flow in a cycle system for a given configuration, cross-sections, and fan/pump pressure.

For cycle network systems Martin and Peters (1963) first used the Newton-Raphson method for water supply systems. Stoecker et al. (1974) for simulating central chilled-water systems and Gregory et al. (1975) for duct systems and was later implemented into a computer code called Tvent1p. Walton (1984) successfully used the Newton method for natural airflow modeling in buildings and managed to increase the converging speed (Airnet computer program).

The use of the Newton method was studied by Lam and Wolla (1972). They analyzed the benefits and shortcomings of the Newton method.

The following three methods are most suitable for solving our problem: Affine invariant Newton method, Tensor method, and Homotopy method. However, the Homotopy method should be considered as the method of choice. Discussion about why we came to this conclusion will be presented after detailed analysis of each method. The method chosen for approximating the Jacobian is very significant for the effective code programming. This discussion will include different techniques for Jacobian calculations. Finally, numerical comparison between different techniques for solving linear equations will be furnished. In last five years numerical solution of linear equations gained a number of new powerful techniques. Now it is possible to reduce the computation time using these new techniques.

Numerical method for nonlinear algebraic equations has a long history. The state-of-art description presented in the excellent books by Rabinowitz (1970), Ortega (1970), Ostrowski (1966), Dennis (1983), and Shnabel (1984). A review of this subject is presented in the excellent book of Nocedal and Wrigth. (1998).

Tenzor methods (Schnabel 1984) are a class of general-purpose methods for solving systems of nonlinear equations. They are specifically intended to solve problems where Jacobian matrix is singular or poorly conditioned, while remaining at least as efficient as standard methods on nonsingular problems. Their distinguishing feature is that they base each iteration on the simple second order term. This term allows it to interpolate more nonlinear functions than standard linear model based methods without significantly increasing the cost of forming, storing or solving the model. There are two types of tensor methods: derivative method that calculates an analytic or finite difference Jacobian at each iteration, and secant method that avoids Jacobian evaluations. Both require no more storage or arithmetic operations per iteration than standard linear model based methods. The attention to this approach has been growing after incorporating in the work (Bouricha 1992).

The tensor method is more robust than the damped Newton method. The tensor method is a special case of the regularization operator approach. (Furi 1969, Tihonov 1986). Theoretically it is possible to find situation where tensor method will be unsuccessful because the regularization operator used sometimes can loose the compactification property. Detailed description of the methods for solving badly posed problems and analyzing regularization operator properties can be found in publication by Tihonov (1986). The tensor method can be applied by experienced users for solving practical problems, but some probability exists that it will fail. Inexperienced user needs a more robust approach.

The Homotopy methods are very robust and capable of achieving global convergence with high probability. The essence of all such algorithms is construction of an appropriate Homotopy map and then tracking some smooth curve in the zero set of this homotopy map. Homotopies are a traditional part of topology, and have found significant application in nonlinear functional analysis and differential geometry. The concepts of homotopy maps, continuation, incremental loading, and invariant imbedding are widely used and intertwined. Parameter continuation is a well established technique in numerical analysis. The basic idea of parameter continuation is to solve a series of problems as some parameter is slowly varied, using a locally convergent iterative technique for each problem, and the solution of the previous problem as the starting point for the current problem. Similar techniques in engineering are known as incremental loading and displacement incrementation.

A fundamental breakthrough occurred with appearance of truly globally convergent simple fixed point algorithms. These algorithms are grounded in topology, constructive, potentially extremely powerful, but horribly inefficient in their early forms. Another significant advantage was the differential equation formulation of continuation, proposed in various forms by Rheinboldt (1982). The next advance was the development in 1976 by S.N.Chow (Chow 1978) of probability one homotopy methods. The thorn of singular Jacobian matrices was finally removed, since these methods were specifically constructed not to have any singular points. The phrase “probability one” refers to the supporting theory, which says that for almost all choices of some parameter vector involved into the homotopy map, there are no singular points and the method is globally convergent. The homotopy algorithm the most robust of all.

Different approaches for solving systems of nonlinear equations were analyzed. It was shown that the more promising approaches are the family of Newton Affine Invariant methods, the Tensor methods and the Homotopy methods.

### *9.4 SPARK evaluation*

The Simulation Problem Analysis and Research Kernel (SPARK) developed by LANL is an object oriented software system that performs simulation of a physical systems by mathematical model, composed of differential and/or algebraic equations. The smallest programming element is a class consisting of an individual equation, called atomic class. Then macro classes bring together several atomic classes (and possibly other macro classes) into a higher level unit. Problem models are similarly described, using the atomic classes, and placed in a problem specification file. When the problem is processed by SPARK, the problem specification file is converted to a C++ program, which gets compiled, linked and executed to solve the problem for given boundary conditions.

During SPARK evaluation the following questions were studied:

1. The class library
   - which airflow elements models are developed
   - physic of the models
   - algorithmic efficiency

2. Method for equations solution
   - theoretic convergence estimation
   - robustness
   - algorithmic efficiency

3. The quality of the interface

Following is the results of analysis of the effectiveness of using SPARK for duct system simulation:

1.  SPARK has a large number of classes developed.  Some of them are simple and used as the basis for developing more sophisticated models (let us name them as level zero classes).   The goal of the framework is to increase the level of abstractions handling by hiding low level abstractions from the user. The framework is overloaded by low level abstractions.  It is not clear for us why the abstractions like "pow" are better then the native C/C++ real data type.

2. The SPARK field is building thermodynamics. All classes from the HVAC Tool Kit are developed for thermodynamic calculation. The airflow calculation task is very specific and needs special abstractions.

3. One of the potential problems for SPARK usage is that many modern ODE solvers work with global sparse matrixes and it is not easy to take them from SPARK data structure.  This seems to be one of the main difficulties of using SPARK to solve duct optimization problems.  SPARK stands aside from the main direction of the methods for ODE solution developing.  Modern solvers use very sophisticated methods for time step control, order control and error control. It is not clear how it is possible to use them during the SPARK calculations.  This is why SPARK data structures are not useful for modern solvers application.  Moreover, considerable additional time is associated with data reformatting during SPARK - ODE solver data exchange.

4. Another important weakness of SPARK occurs when one tries to solve problems connected with a system of nonlinear equations.  One of the main components of the programs from this field is the linear equations solvers. The state of the art in this field is LAPACK. Subroutines from this package are optimized for specific computers. Without doubt, dense linear algebra subroutines must come from the LAPACK.

5. "Open implementation" (one of the most interesting and useful modern technique) seems to be out of SPARK concept. It is not clear how "open implementation" concept can be incorporated into SPARK. On the other hand, ideas enclosed in "open implementation" can be extremely useful for complicated applications.   Software has traditionally been constructed according to the principle that a module should expose its functionality but hide its implementation. This principle, informally known as black-box abstraction, is a basic tenet of software design, underlying our approaches to portability, reuse, and many other important issues in computing. However, exposing only the functionality of a module in its interface can sometimes lead to performance difficulties when the module gets reused. It has been observed that in such cases, clients "code around" the problem by either re-implementing an appropriate version of the module or by using existing modules in contorted ways. Many recent systems address this problem by having modules that allow client control of their implementation strategy. The open implementation approach works by somewhat shifting the black-box guidelines for module design (Kiczales 1996). Whereas black box modules hide all aspects of their implementation, open implementation modules allow clients some control over selection of their implementation strategy, while still hiding many true details of their implementation. In doing this, open implementation module designs strive for an appropriate balance between preserving the kind of opacity black-box modules have, and providing the kind of performance tailorability some clients require. Kiczales (1996) presents a simple but clear example for motivation for the "open implementation". Consider the two class definitions:

```
class position
{// data
 int x, int y;
   // methods
 ……..
  public: position (int a,int b):  x(a),y(b){}
};
```

```
class person
{ // data
NAME  name, age, address,…..;
// methods
……………………….
public: person(….)
};
```

The class 'position' might be part of a graphic application, where the instances are used to represent the position of the mouse as it moves. The class defines two slots x and y. In this case, the behavior of the program is such that there will be a very large number of instances, both slots will be used in every instance and access to those slots should be as fast as possible. The second class 'person' might come from a knowledge representation system, where the instances are being used as frames to represent different individuals. The thousand slots defined in the class correspond to a thousand properties of a person that might be known. In this application the behavior is such that although there will be a very large number of instances, in any given instance only a few slots will actually be used. Furthermore, access to these properties will rarely be in the inner loop of a computation. Clearly, the ideal instance implementation strategy is different for the two classes. For 'position', an array-like strategy would be ideal; it provides compact storage of instances, and rapid access to the x and y slots. For 'person', a hash-table-like strategy would be more appropriate, since it isn't worth allocating space for a slot until it is known that it will be used. This makes access slower, but it is a worthwhile tradeoff given a large number of instances.
The likely default implementation in most object-oriented languages is the array-like strategy. This serves the author of the 'position' class quite well, but the author of the 'person' will not be so happy. Even though the language abstraction serves to express their program quite clearly, hidden properties of the implementation will impair performance.

6. SPARK input language seems to be time consuming and error prone. It is still text based. From the other point of view SPARK input language semantic is very clear and can be easily transformed into 3D visualization. This can drastically change the SPARK environment. As an example, OpenGL is designed to support interactive applications by providing a selection mechanism that automatically tells which objects are drawn inside a specified region of the window. It is possible use this mechanism together with a special utility routine to determine which object within the region is specified by the user or picked with a cursor. Moreover, hierarchical model of picking can easily be applied (128 of hierarchical levels under WINDOWS NT 4.0 is possible to handle). Another benefit from GUI is the possibility to hide from the user small details of connections between different objects.

7. The main idea used in SPARK, which is that of problem reducing techniques based on graph theory, can be embedded into the Multilevel Splitting approach. The difference is that SPARK uses two different levels (strong connected component as level 1 and complete simulation model as level 2). The so-called multilevel additive Schwarz method (Shao 1993) is similar to the method used by SPARK**.** Its idea is to use a hierarchy of levels for a multiscale representation of the problem and to combine the contributions of all levels as a sum. This process implicitly defines an operator sum that is well behaved and that has bounded condition number independent of the number of levels. Thus, it is suitable for fast iterative inversion by conjugate gradient algorithms. The version of SPARK under revision has no implementation of conjugate algorithm, but it seems that modernization can be easily elaborated. The recent theoretical approach to these methods by Oswald is based on results in approximation theory, in particular on methods from the theory of Besov spaces. The relevant basic results can be found in (Berg 1976).

From a more general perspective, the multilevel additive Schwarz method is also related to multigrid methods and their theory. Classical multigrid methods can be interpreted as a multiplicative Schwarz method where the levels are visited sequentially and the basic structure is a product of operators. Refer to Hackbuch (1986) and McCormick (1992) for the classical theory and to Yserentant (1990) for a review of recent developments.

In the last decade, a new class of multilevel approach based iterative methods for the numerical solution of partial differential equations has emerged: the multigrid methods and domain decomposition methods. These methods use the properties of the underlying differential equation to obtain fast convergence of the iterations at a small cost, but they are difficult to apply to complicated, real-word problems and they require user expertise and guidance for good performance. The domain decomposition methods are very close to by component splitting technique used in SPARK. Main ideas from this field are given below. For simplicity, we restrict ourselves to the solution of a system of linear equations.

$$Ax=f$$

with a symmetric, positive definite matrix A. The method of choice is then the preconditioned conjugate gradients method (Golub 1989).

**Step 1**. Given $x_0=0$, let k=0 and

$$r_0=b-Ax_0=b$$

$$h_0=Cr_0$$

$$p_0=h_0$$

**Step 2.** Do

$$\boldsymbol{a}_k = \frac{r_k^T p_k}{(A p_k)^T p_k}$$

$$x_{k+1} = x_k + \boldsymbol{a}_k p_k$$

$$r_{k+1} = r_k - \boldsymbol{a}_k A p_k$$

$$h_{k+1} = C r_{k+1}$$

$$\boldsymbol{b}_{k+1} = \frac{r_{k+1}^T h_{k+1}}{r_k^T h_k}$$

$$p_{k+1} = h_{k+1} + \boldsymbol{b}_k p_k$$

**Step 3**. If $r_k$ is small enough, exit; else k=k+1 and go to Step 2. In each step, this method invokes the evaluation of the matrix-vector products Ax and Cr. The number of steps to achieve a given precision in the energy norm $\|x\|_A = \sqrt{x^T Ax}$ is at most proportional to $\sqrt{k}$, where k is the condition number

$k = k(CA) = l_{\max}(CA) \,/\, l_{\min}(CA)$ .

where $l_{\max}$ and $l_{\min}$ denote the smallest and the largest eigenvalues, respectively. In practice, the number of iterations is often much lower, but this luck tends to disappear with growing size of the problem and with better preconditioning that decreases the condition number. The task of designing an efficient preconditioned conjugate gradient method is thus one of finding a pre-conditioner C such that k is small and the matrix-vector product Cr is inexpensive to evaluate. One of the most efficient pre-conditioners is built by the abstract additive Schwarz method (Bjorstad 1991).

Let V be some space of degrees of freedom, so that the matrix A is understood to be a linear mapping A: V -> V. Let $V_0, \ldots, V_m$ be subspaces of the space V such that

$$V = V_0 + \ldots + V_m. \qquad (1)$$

The action of pre-conditioner C is defined by solving this problem on each subspace separately and adding the results to compute u=Cr:

$$u_i \in V_i : \quad v_i^T A u_i = v_i^T r, \ \forall v_i \in V_i, \ i = 0,\ldots,m, \qquad (2)$$

$$u = \sum_{i=0}^{m} u_i \qquad (3)$$

The method is implemented using a linear operator M: $V_0 \times \ldots \times V_m \to V$ which leads to representation

$$Cr = MD^{-1}M^{T}r \qquad (4)$$

The operator M corresponds to the summation (3), including a possible change of basis, since the subspaces $V_I$ need not be spanned by subsets of the shape functions that span V. The matrix D is the block diagonal of $M^{T}AM$ corresponding to the decomposition (1), and the action of $D^{-1}$ corresponds to the simulation solution of the subproblems (2) on the spaces $V_i$. The operator M is not stored as a matrix but rather in a factored form as a recording of the multipliers in the change of basis.

A specific preconditioning strategy now consists in selecting subspaces $V_i$. In an intelligent block preconditioning method, this selection is fully automatic and guided by numerical estimators that attempt to ensure an acceptable condition number while keeping the dimension of the spaces $V_I$ as small as possible to archive a low computational cost per iteration. Numerous other preconditioning methods are of the present form, with the matrix D possibly replaced by an approximation that is cheaper to invert and with various selections of the spaces $V_I$ (Dryja 1992).

The key observation here is the P.L.Lions Lemma (Dryja 1990), which states that if there is a constant $c_0$ such that every $v \in V$ has a decomposition satisfying:

$$v = \sum_{i=0}^{m} v_i, \quad v_i \in V_i, \quad \sum_{i=0}^{m} \|v_i\|_A^2 \le c_0 \|v\|_A^2 \qquad (5)$$

then $l_{\min}(CA) \ge 1/c_0$ . The maximum eigenvalue is easily bounded as the maximum number of the spaces $V_I$ that have a nontrivial intersection:

$$l_{\max}(CA) \le \max_{i=1,\ldots,m} \left|\{V_j : V_j \cap V_i \ne \{0\}\}\right|. \qquad (6)$$

In other words, the condition number bound depends on boundedness in energy of decomposition (1) and on connectivity of overlaps of the spaces $V_i$.

In general, the space $V_0$ is selected to be a coarse space that represents the whole problem approximately, while the spaces $V_1, \ldots, V_m$ represent substructures or overlapping subdomains. Usually it is then possible to prove that the condition number is bounded independently of the number of substructures. If the null-space of the principal terms of the differential operator can be represented exactly in the coarse space $V_0$, then it can be proved that the constant $c_0$ depends on local properties of the substructures only. The general philosophy is to first select spaces $V_I$ that are known to work well and for which asymptotically optimal theoretical bounds on the energy of the decomposition can be proved by the methods of Functional Analysis. Then numerical estimators are to be used to decide when to merge the spaces $V_I$ into larger spaces.

The approach described above works well only for comparatively good numerical models. An important condition is having small differences in global matrix coefficients (for hydraulic net this mean equal hydraulic characteristics of the volumes). In reality the difference between hydraulic volumes is significant. For this reason we have to deal with a particularly challenging class of problems, which arises with models described by partial differential equations (PDE) with discontinuous coefficients.

There are several difficulties associated with the numerical solution of these problems using preconditioned iterative methods. For a number of methods, the larger the jumps of the coefficients, the slower the convergence. However, it has recently been shown (Bakhvalov 1991) that for continuous models and with a special initial guess, the rate of convergence does not depend on the size of the jumps. The authors consider a parametric family of symmetric matrices $A_w = A_0 + wB$ with a nonnegative definite matrix $A_0$, a positive definite matrix B, and a parameter $w, 0 \le w \le 1.$ Small $w$ leads to a large drop of coefficients of $A_w$. For solving linear algebraic equations with matrix $A_w$ ,authors use standard preconditioned iterative methods with matrix B as a pre-conditioner. It is known that a proper choice of the initial guess makes it possible to keep all residuals in the subspace $\mathrm{Im}(A_0)$ and the difference between all iteration vectors and the solution in the subspace $\mathrm{Im}(B^{-1}A_0)$.

Another similar direction in the numerical methods is Parallel Multisplitting. A problem from this field can be described by one of the following equations:

$$f(x)=0 \qquad (7)$$

$$\min_{x \in D} f(x) \qquad (8)$$

Here in (7), f(x) can refer to a linear equations, Ax=b, or nonlinear equations for $x \in R^n$, $f : R^n \to R^m, m \le n.$ In (8), f(x) may be a nonlinear functional for which a minimum is required over a subset $D \subseteq R^n$, or over-determined linear equations for which a least squares solution is sought.

Multisplitting for Ax=b: A multisplitting of A is defined as follows:

***Definition 1***. Given a matrix $A \in R^{n \times n}$ and a collection of matrices $M^j, N^j, E^j \in R^{n \times n}, j = 1 : p,$ satisfying

(i) A=M$^j$-N$^j$ for each j, j=1:p,
(ii) M$^j$ is nonsingular, j=1:p,
(iii) E$^j$ is a nonnegative diagonal matrix, j=1:p and $\sum_{j=1}^{p} E^j = I$.

Then the collection of triples $(M^j, N^j, E^j), j = 1 : p$ is called a multisplitting of A and the LMS method is defined by the iteration:

$$x^{k+1} = \sum_{j=1}^{p} E^j (M^j)^{-1} (N^j x^k + b), \; k = 1,.... \qquad (9)$$

The advantage of this method is that at each iteration there are p independent iterations of the kind

$$M^j y_j^k = N^j x^k + b, \; j = 1 : p, \qquad (10)$$

where $y_j^k$ represents the solution to the local problem. Hence the work for each equation in (10) is assigned to one processor and communication is required only to produce the update given in (9). In general most of the diagonal equations of the diagonal elements in E$^j$ have entry values of only 0 and 1. Otherwise the algorithm uses overlap and optimal overlap has to be determined. Convergence of the lLMS method for various A, i.e., A an M matrix, an H matrix, symmetric positive definite, has been investigated by several researchers (Frommer 1989). Relaxed multisplitting depending on a parameter ù is used to accelerate convergence and has also been considered. Efficient new algorithms by Huang and O'Leary [Huang 1994] use multisplitting for GMRES, conjugate gradients and other iterations.

Nonlinear splitting for f(x)=0
Consider the solution of (5), m=n, by a Newton method. A search direction d is found as the solution of the linear system

$$\nabla f(x^k) d + f(x^k) = 0. \qquad (11)$$

Hence a solution can be found by applying a LMS to the matrix $\nabla f(x)$. Alternatively, the splitting can be applied directly to f(x):

***Definition 2. Nonlinear multisplitting.*** For j=1:p let $F^j : R^n \times R^n \rightarrow R^m$ be such that $F^j(x, x) = f(x)$ for all $x \in R^n$, and E$^j$ as in Definition (1). Then the collection of pairs $(F^j, E^j), j = 1 : p,$ is called a nonlinear multisplitting of f and the NLMS method is defined by the iteration:

$$x^{k+1} = \sum_{j=1}^{p} E^j y^{k,j}, k = 0,1,... \qquad (12)$$

where $y^{k,j}$ solves $F^j(x^k, y^{k,j}) = 0.$

This is a natural extension of Definition (1) since the LMS method for solving f(x)=Ax-b can be split in the nonlinear sense by taking $F^j(x,y) = M^j y - N^j x - b.$ For any nonlinear function this can be decomposed as $f(x) = \sum_{j=1}^{n} F_j(x)$. An obvious NLMS is obtained by taking $F^j(x,y) = F_j(y) + \sum_{i \neq j} F_i(x)$ and $E^j = \frac{1}{n} I$. This leads to an iterative scheme of "alternative direction" or Peaceman-Rachford" type [Ortega 1970]. This approach does not necessarily produce subproblems of reduced size since the Newton method for each $F^j(x,y) = 0$ requires "inversion" of a Jacobian matrix for $F^j(x,y)$ of order n, unless $E^j_{ii} = 0$ for some set of $i, 1 \le i \le n.$ To reduce the subproblem size suppose that (1) is solved with respect to several possibly overlapping blocks of variables.

***Definition 3. Nonlinear block.*** The increasing availability of advanced-architecture computers is having a very significant effect on all spheres of scientific computation, including algorithm research and software development in numerical linear algebra. The time of coming multiprocessor personal computers for HVAC users seems to be near. Linear algebra, and in particular solution of linear systems of equations, lies at the heart of most calculations in scientific computing. For the past 15 years there has been a great deal of activity in the area of algorithms and software for solving linear algebra problems (Dongarra 1994). The linear algebra community has long recognized the need for help in developing algorithms into software libraries, and several years ago as a community effort put together a de facto standard for identifying basic operations required in linear algebra algorithms and software. The hope was that the routines making up the standard (known collectively as the Basic Linear Algebra Subprograms (BLAS)) would be efficiently implemented on advanced-architecture computers by many manufactures, making it possible to reap the portability benefits of having them efficiently implemented on a wide range of machines.

Over the past years several important packages of dense linear algebra software were developed: LAPACK++ and BLAS++. LAPACK++ is an object-oriented C++ extension to the LAPACK library for numerical linear algebra. This package includes state of the art numerical algorithms for the more common linear algebra problems encountered in scientific and engineering applications: solving linear equations, linear least squares, and eigenvalue problems for dense and banded systems. LAPACK++ provides the speed and efficiency competitive with native Fortran codes while allowing programmers to capitalize on the software engineering benefits of object-oriented programming. The design goals of LAPACK++ include:

- Maintaining performance competitive with Fortran.
- Providing a simple interface that hides implementation details of various matrix storage schemes and their corresponding factorization structures.
- Providing a universal interface and open system design for integration into user-defined data structures and third-party matrix packages.
- Replacing static work array limitations of Fortran with more flexible and type-safe dynamic memory allocation schemes.
- Providing an efficient indexing scheme for matrix elements that has minimal overhead and can be optimized for in most application code loops.
- Utilizing function and operator overloading in C++ to simplify and reduce the number of interface entry points to LAPACK.
- Utilizing exception error handling in C++ for intelligent managing of error situations without cluttering up application codes.
- Providing the capability to access submatrices by reference, rather than by value, and perform

factorization "in place". This is vital for implementing blocked algorithms efficiently.

- Providing more meaningful naming conventions for variables and function names. (Names no longer limited to six alphanumeric characters).
- LAPACK++ also provides an object-oriented interface to the Basic Linear Algebra Subprograms (BLAS) allowing programmers to utilize these optimized computational kernels in their owns C++ applications.

***Basic linear algebra subroutines (BLAS++)***

The Basic Linear Algebra Subroutines (BLAS) have been the key to obtain good performance on a wide variety of computer architectures. The BLAS define a common interface to low-level operations often found in computational kernels. These operations such as matrix/matrix multiplication and triangular matrix solves typically comprise most of the computational workload found in dense and banded linear algebra algorithms. The Lec=vel 3 BLAS obtains good performance on a wide variety of architectures by keeping data used most often in the closest level of memory hierarchy (register, cash, etc.). The BLAS++ interface simplifies many of the calling sequences to the traditional f77 BLAS interface, by using the LAPACK++ matrix classes. These routines are called within the LAPACK++ algorithms, or can be called directly at the user-level within applications. There are two levels of the BLAS++ interface. The first is a direct interface, and is essentially unlined to call BLAS directly in eliminating any overhead. The other, more elegant interface overloads the binary operators * and + for simple expressions such as C=A*B. Having these two interfaces gives the user the choice between simplicity and performance in their application codes.

Unlike dense linear algebra, sparse linear algebra is still under extensive development. One very important field of linear algebra which can drastically increase the size of the numerical models handling is not reflected in SPARK. It is Krylov's subspace algorithms for system of equations solution. Although successive over-relaxation methods and the first Krylov subspace methods were developed at the same period of time (end of sixties), the class of Successive Over-relaxation methods became the method of choice, since they require a small amount of computer storage. These methods were quite well understood by Young (Young 1971) and the theory was covered in great details in Varga's book (Varga 1962). It may seem evident that iterative methods gained in importance as scientific modeling let to larger problems, since direct methods are often too expensive in terms of computer memory and CPU-time requirements. The past ten years led to well-established and popular methods, that all lead to the construction of approximate solutions in the so-called Krylov subspace. Given a linear system Ax=b, with a large, usually sparse, asymmetric nonsingular matrix A, then the standard Richardson iteration

$$x_k = (I - A)\,x_{k-1} + b$$

generates approximate solutions in shifted Krylov's subspaces

$$x_0 + K^k(A; r_0) = x_0 + \{r_0, A r_0, ..., A^{k-1} r_0\},$$

with $r_0 = b - A x_0$, for some given initial vector $x_0$.

The Krylov subspace projection methods fall into three different classes:

1. The Ritz-Galerkin approach: Construct the $x_k$ for which the residual is orthogonal to the current subspace:

$b - A x_k \perp K^k (A; r_0).$

2. The minimum residual approach: Identify the $x_k$ for which the Euclidean norm $\|b - A x_k\|_2$ is minimal over $K^k$ $(A;r_0)$.

3. The Petrov-Galerkin approach: Find an $x_k$ so that the residual b-A$x_k$ is orthogonal to some other suitable k-dimension subspace.

The Ritz-Galerkin approach leads to such popular and well-known methods as Conjugate Gradients, the Lanczos method, DFOM, and GENCC=G. The minimum residual approach leads to methods like GMRES, MINRES, and ORTHODIR. If we select the k-dimensional subspace in the third approach as $K^k(A^T;s_0)$, then we obtain the Bi-CG, and QMR methods. More recently, hybrids of the three approaches have been proposed, like CGS, Bi-CGSTAB, BiCGSTAB(l), TFQMR, FGMRES, and GMRESR.
Most of these methods have been proposed in the last ten years. GMRES is the most robust of them, but in terms of work per iteration step it is also the most expensive.

### *Differential-algebraic equations*

This is a very big and well developed field. The most powerful algorithms from this field should be utilized in such challenging tasks as building modeling. One of the most important branches of numerical ODE is a numerical solution of systems of differential-algebraic equations (DAEs) (Brenan 1995). These systems can be found in a wide variety of scientific and engineering applications, including circuit analysis, computer-aided design and real-time simulation of mechanical systems, power subsystems, chemical process simulation, and optimal control. During the same period of time much work has also been done in the field of numerical solution of delay differential equations (DDEs) (Baker 1995). Delay differential equations arise from, for example, real-time simulation, where time delays can be introduced by the computer time needed to compute an output after the input has been sampled, and where additional delays can be introduced by the operator-in-the-loop. Delays arise also in circuit simulation and power systems, due to, for example, interconnections for computer chips and transmission lines, and in chemical process simulation when modeling pipe flows.

Delay differential algebraic equations (DDAEs), which have both delay and algebraic constraints, appear frequently in these fields. Differential-algebraic equation system in its general form is

$$F(t,y,y')=0, \; v(0)=v_0.$$

where F, y, and y' are N-dimensional vectors. The DAE system must be index-one. For semi-explicit DAE

systems (ODEs coupled with nonlinear constraints) of the form

$$v_1^{'} = f_1(v_1, v_2, t)$$

$$0 = f_2(v_1, v_2, t),$$

the system is index-one if $\frac{\partial f_2}{\partial v_2}$ is nonsingular in a neighborhood of the solution. The initial condition given to solver must always be consistent. For semi-explicit DAE systems this means that the initial conditions must satisfy the constraints. Given a consistent set of initial conditions, DAE is solved over the given time interval via an implicit, adaptive-stepsize variable order numerical method. The dependent variables and their derivatives are discreetized via backward differentiation formulas (BDF) of orders one through five. At each time step this yields a nonlinear system that is solved using a modified Newton iteration. The linear system at each Newton iteration is solved via either a dense or banded or sparse direct linear system solver. For large-scale DAE systems it is possible to solve the nonlinear system at each time step using an inexact Newton method. This means that the linear systems at each iteration are not necessary solved exactly. In fact, they are solved approximately via a preconditioned iterative method.

Very important field for airflow optimization is a combinatorial and mixed optimization. Combinatorial optimization is involved with models and methods for optimization over discrete choices. It is routed in the theory of linear programming, and has strong links with discrete mathematics, probability theory, algorithmic computer science, and complexity theory. Some problems in the area are relatively well understood and allow solution to optimality in polynomial time. Many others are NP-hard, and one is forced to choose one of three ways: an enumerative method that is guaranteed to produce an optimal solution, an approximation algorithm that runs in polynomial time, or some type of heuristic search technique, without any a priori guarantee in terms of solution quality or running time.

Branch-and-bound remains the most commonly used technique for solving (mixed) integer programs. The quality of the available upper and lower bounds on the optimal value of the considered instance is the decisive factor for success of this tree search technique.

***Conclusions***

I am impressed by the work and skills involved in SPARK development.   SPARK framework is based on modern method for solution of the nonlinear system of equations and has a large set of useful numeric thermodynamic models for different types of building equipment handling.  All SPARK models well developed and documented.

For the time being, the SPARK's main field is the thermodynamic calculations.  SPARK needs significant expansion for task of the ventilation system simulation and optimization handling.

The main directions for expansion are:

1. The HVAC Toolkit must be significantly increased in size. As an example a numerical model for several hundreds of local hydraulic resistances must be included.
2. Numerical methods for sparse matrices handling must be included. The developed numerical methods for matrix calculation have so far been applied only to the dense matrixes.
3. Dense matrixes calculations must be based on LINPACK library (or on the libraries from the next generations like LAPACK, LAPACK++). Algorithms from this library are supported and tuned to specific computers. Any other choice will have result in decreasing numerical efficiency.
4. Modern numerical methods for systems of nonlinear equations solution with large number of variables must be included in the SPARK. At any rate, the SPARK environment must have support for such type of algorithms.
5. Mathematics of the methods based on numerical model decomposition /splitting  (by graph algorithms, by hands, or by neuron nets as in several modern works) is comparatively complicated and still under

development. The user must have a possibility for modernization of SPARK algorithms and for inserting new algorithms.

6. The SPARK input files have a large size for comparatively small models. Models having several hundreds of elements can be very hard to develop. It is seems that a good script language can be useful.
7. SPARK visual environment has not used the comparatively new 3D possibilities. SPARK input is still text-based. DirectX- or OpenGL-based user interface can be extremely useful.
8. The connection SPARK - GENOPT seems to be unpractical, at least for now. SPARK classes have a potential for calculating derivatives. Newton-like algorithms can be easily applied to the SPARK so in the future SPARK models can be optimized by the most effective algorithms.
9. The relational extension for C++ can be extremely useful for Framework construction. As an example, conditions applied to the input/output parameters can be inserted on the declarative level.
10. It seems that application of ports and links to abstract data types can be a great improvement for SPARK and can save a lot of time during numerical model development in SPARK environment. This can by developed comparatively easy on the preprocessor level. Another useful abstraction would be the "port" - "class" adapter (e.g. adapters used in STL (Stepanov 1995)).
11. Some SPARK models differ only in implementation details. It could be useful to reduce the number of classes while keeping the functionality by using open implementation concepts (Kiczales, 1996). As a benefit, a multilayer class library design would become possible (Klir 1990).

### *9.5 GenOpt evaluation*

Optimal design of ventilation systems is a complex mathematical problem. The following list presents several requirements associated with such types of problems:

1. The time required to complete a single function evaluation with one parameter set is long. Hence, minimization of the number of function evaluation is vital.
2. Analytic derivatives (with respect to parameters) of the objective and constraint functions are frequently unavailable. Hence, sensitivity-based optimization methods depend upon numerically generated gradients which require additional function evaluations for each scalar parameter,
3. The parameters may be either continuous or discrete, or a combination of the two,
4. The objective and constraint functions may not be smooth or well behaved; i.e., the response surfaces can be severely nonlinear, discontinuous, or even undefined in some regions of the parameter space. The existence of several local extremes (multimodality) is common,
5. Convergence tolerances in embedded iteration schemes introduce nonsmoothness in the functions evaluation response surface, which can result in inaccurate numerical gradients,
6. Each function evaluation may require an "initial guess". Function evaluation dependence on the initial guess can cause additional nonsmoothness in the response surface. Moreover, a solution may not be attainable for an inadequate initial guess, which can restrict the size of the allowable parameter changes.

Each of the listed challenges can adversely affect the robustness and efficiency of optimization processes and has led to the research and development of application-specific techniques, new optimization algorithms, multilevel hybrid and sequential approximate optimization strategies, automatic differentiation and adjoint augmentation methods. When variables are discrete a concept or a derivative break down and recourse must be taken to nongradient-based combinatorial methods. Discrete and mixed continuous-discrete problems can be tackled with genetic algorithms, branch and bound, simulated annealing, and many other nongradient-base approaches.

The field for GenOpt application is a set of problems for which Newton-based methods are inappropriate or inapplicable. It is usually the problem of minimization of a nonlinear function **f** with one or more of following properties:

1. Calculation of **f** is very expensive or time consuming. For example, each value of **f** may be obtained by solving a costly numerical subproblem.
2. Exact first partial derivatives of **f** cannot be calculated. Either the gradient does not exist - for example, when **f** is unpredictably discontinuous - or **f** is defined from a complex or convoluted computational structure that obviates application of automatic differentiation techniques.
3. Numerical approximation of the gradient of **f** is impracticably expensive or slow.
4. The values of **f** are "noisy". For example, the calculated value of **f** may depend on discretization, sampling on a grid, inaccurate data, or an adaptively solved subcalculation.

In many of the practical problems where such functions occur, a highly accurate solution is neither possible nor desired: it may be impossible because of uncertainties and errors in the underlying model or data, or may be undesirable because of the unacceptably high cost required to attain it. In fact, a frequent aim in these applications is improvement rather than optimization. Thus the critical issue is finding a better answer quickly.

The best choice for minimizing or improving a function with the properties listed above, is a direct search method (Gill 1983). Since its publication in 1965 (Nelder 1965), the Nelder-Mead algorithm has been used in an extraordinarily wide variety of contexts. Despite its wide use, until recently the Nelder-Mead method has been deprecated or ignored by almost all of the optimization community. This negative attitude arises in part as a reaction to users who might choose a Nelder-Mead method simply for its ease of use, even when minimizing a smooth and inexpensive function well suited to a more reliable gradient-based method. The first theoretical results concerning the Nelder-Mead method appeared in the 1985 thesis of Woods (Ph.D. of Woods 1985). Woods provides a negative result by depicting a two-dimensional nonconvex function for which Nelder-Mead iteration is a shrink and all vertices converge to a non-minimizing point. Woods then considers strictly convex functions with bounded level sets, and defines a modified Nelder-Mead method with a stricter descent requirement for accepting the reflected point during the calculation steps. Under these assumptions Woods proves that:

(1) Every convergent subsequence of the simplexes generated by the modified algorithm converges to a degenerate simplex (a single point),
(2) The values of **f** and all limit points are equal,
(3) The set of limit points is connected,
(4) The proofs do not apply to the original Nelder-Mead method; convergence of the simplexes to a single point is not guaranteed, but only convergence of a subsequence.

A renaissance of interest in direct search methods began in 1989 with Torczon's Ph. D. thesis (Torczon 1989). Her thesis and subsequent papers propose a new form of direct search method - the multidirectional search.

The multidirectional search method is simplex-based, and consequently draws on ideas explained by Nelder (Nelder 1965). Each iteration is associated with a current simplex with the best vertex (the vertex with the lowest function value). Operations called reflection, expansion, and contraction are the same as those in the Nelder-Mead method, but this does not involve the **n** edges of the simplex emanating from the best vertex, so that the entire simplex is reflected, expanded, and contracted. A multidirectional search iteration succeeds when it finds a point of strict improvement over the best vertex, in contrast to the much weaker condition in a Nelder-Mead iteration of finding a strict improvement compared to the worst point. However, according to Nelder-Mead, the acceptance criterion in multidirectional search is only decreasing. A multidirectional search method can require substantially more function evaluations than the Nelder-Mead method presents. Typical Nelder-Mead iteration requires either one or two function evaluations. The number of function evaluations required by a multidirectional search iteration is an integer multiplied by **2n,** where **n** is the number of parameters (unknowns). A great advantage of the multidirectional search method is its strong convergence properties, which have been refined and extended by Torczon to include all pattern search methods. The key ingredients in Torczon's convergence proof are the uniformed linear independence of the simplex edges at every iteration, the scale d lattice structure of all generated points, and the step control strategy. These features together ensure that pathological simplexes and steps cannot arise in pattern search methods. Torczon's general theory applies to coordinate search, evolutionary design, the Hooke-Jeeves method and multidirectional search but not to the Nelder-Mead method.

Torczon made a very important investigation of the Nelder-Mead method: the selection of an initial simplex. Nelder and Mead placed no restrictions, other then nondegeneracy, on the shape of the initial simplex. Most sources simply note this mild restriction but do not suggest ways to generate the initial simplex from a given initial estimate. For instance, Jacoby, Kowalik and Pizzo (Jacoby 1972) note that the Nelder-Mead’s simplex algorithm only requires a general simplex, but they suggest that the construction of a regular initial simplex assures that its vertices span the full space. Parkinson and Hutchinson (Parkinson 1972) suggest to start the Nelder-Mead simplex algorithm with a right-angled simplex. This simplex can be generated by defining each of the new vertices to be at some fixed distance from the initial guess for each of the **n** coordinate directions. Parkinson and Huchinson notice that in their investigation of the efficiency of the Nelder-Mead simplex algorithm, the shape of the initial simplex proved to be relatively unimportant. This result was expected, since the initial simplex is rapidly modified by the action of the algorithm. Moreover, the shape of the simplex is scale dependent, so that usually a regular simplex may not be a desirable choice.

### *The selection of the initial size*

While the size of the initial simplex definitely affects performance, the scale dependency of the simplex creates general guidelines as to the best size for the initial simplex. Particularly when the simplex is too large, the problem is that the algorithm may spend a significant number of iterations simply contracting the simplex before it can make any real progress. Theoretically the Nelder-Mead simplex algorithm only rescales the entire simplex as a last resort. In this case it takes a "shrink" step. In reality, the shrink step is almost never taken. Instead the Nelder-Mead simplex algorithm changes the size of the simplex by moving only one vertex at a time; however, each change also distorts the shape of the initial simplex. After **n** iterations the size of the simplex may be smaller or larger, but its shape may also be significantly altered in the process.

Parkinson and Hutchinson experimented with changing the size of the initial simplex used to start the Nelder-Mead simplex algorithm. They concluded that varying the size of the initial simplex did produce significant variations in the number of function evaluations required to solve each problem. Consequently, they suggested two strategies for determining the initial size of the simplex. Both strategies require line searches to determine the lengths of the edges in the simplex. Parkinson and Hutchinson reported modest decreases in the number of function evaluations required to minimize the function when either of these strategies were used to determine the size of the initial simplex. The trade-off is that more work, in the form of local line searches, must be done before starting the algorithm.

### *Orientation*

It is well known that the orientation of the simplex affects the progress of Nelder-Mead simplex algorithm. The reason is simple: the orientation determines the search directions. Parkinson and Hutchinson (1972) found that varying the orientation of the initial simplex had a dramatic effect for all the functions they tested using the Nelder-Mead simplex algorithm. They restricted their attention to problems with only two variables and then rotated the simplex about the best vertex in increments of 1 . They found that in some cases a rotation of only 1 changed the number of function evaluations by 45%. Their conclusion was that the deliberate choice of an initial orientation holds an element of risk for all but very regular functions, due to a wide variations in the number of required function evaluations necessitated by even small changes in orientation. The difficulty is compounded by raising the dimensionality of the objective function since this widens the choice of orientation parameters in direct proportion. Parkinson and Hutchinson also tested several automatic procedures which would determine appropriate initial parameters for the orientation of the simplex, but reported that none gave sufficient and regular gains over random selection to merit normal use. One suggestion they did make was that the function values at the vertices of the original simplex be used to approximate the direction of steepest descent. This information could then be used to adjust the orientation of the initial simples so that the initial step would be along this approximation of the steepest descent direction. The difficulty of this approach, as they note, is that the quality of this estimate depends on the scale of the initial simplex.

### *Choosing the scaling factors*

Nelder and Mead tested a variety of choices on three small (n=2, 3, and 4), but difficult problems. Their conclusion was that the simple strategy of either doubling or halving the step sizes was clearly the best. Parkinson and Hutchinson noted that the recommendations made by Nelder and Mead were based on trials with only about one hundred combinations, so they systematically investigated several thousand combinations. They concluded that there was no general strategy which gave the best results for all the test functions. For the test functions they studied - which were again, for the most part, small, complex and limited in number - taking expansion steps of 2.5 and contraction steps of 0.25 proved to be somewhat superior to the values suggested by Nelder and Mead. Walmsley (Walmsley 1981) suggested retaining the expansion factor of 2 while setting the contraction factor to 3/4 in an effort to slow the drastic contractions that often lead to difficulties with the Nelder-Mead simplex algorithm. As the above discussion indicates, the optimal choice of scale factors is as dependent on the function to be minimized as is the choice of the initial simplex. In fact, the two can clearly interact. This makes the choice of optimal scaling factors all but impossible.

### *Stopping criteria*

Nelder and Mead suggested comparing the standard deviation of the function values in the simplex with a preset value and stopping when the standard deviation falls below this value. This leads to the stopping test:

$$\sqrt{\sum_{i=0}^{n} \frac{\left(f\left(v_i^k\right) - \overline{f}\right)^2}{n}} < \boldsymbol{e},$$

where $\overline{f}$ is the mean of the function values at the **n+1** vertices and **å**is the preset tolerance. As Nelder and Mead observed, the success of this criterion depends on the simplex not becoming too small in relation to the curvature of the surface until the final minimum is reached.

Woods (Woods 1985) showed how this stopping criterion can lead to the premature termination. In both his illustrations the difficulty lies in the fact that while the function values at the **n+1** vertices in the simplex is "close", the simplex itself is still relatively large. Thus the algorithm halts without recognizing that the simplex has not yet identified a solution. Parkinson and Hutchinson suggested that the stopping criteria should restrict both the range in **f** and the corrections to $\mathbf{v}_\mathbf{I}$ for all **i**. The proposed the following two tests:

$$f\left(v_n^k\right) - f\left(v_0^k\right) < \boldsymbol{e},$$

where $v_n^k$ is the worst vertex in the simplex, i.e., the vertex with the largest function value, and

$$\frac{1}{n}\sum_{i=0}^{n}\left\|v_i^{k+1} - v_i^k\right\|^2 < \boldsymbol{e}$$

Woods interpreted the second stopping criterion proposed by Parkinson and Hutchinson as a measure of how far the simplex has moved. He noted that the distance the simplex moves is related to the size of the simplex, and on the basis of this observation proposed that the size of the simplex be used as a stopping criterion. This leads to the following test:

$$\frac{1}{\Delta}\max_{1<i<n}\left\|v_i^k - v_0^k\right\| \le \boldsymbol{e} ,$$

where $\Delta = \max\left(1, \left\|v_0^k\right\|\right)$. This would then measure the relative size of the simplex by considering the length of the longest edge adjacent to $v_0^k$.

The performance of the Nelder-Mead simplex algorithm has been tested extensively because it was widely used for many years. Good examples are shown by Torczon (1989). Torczon made a series of tests for six different functions (Penalty, Extended Powel Singular Function, Extended Rosenbrock Function, Trigonometric Function, Variably Dimensioned Function, The Perfect Function ($\mathbf{l}^\mathbf{2}$ norm)). According to Torczon the Nelder-Mead, simplex algorithm is simply not robust. Torczon reported that on every test problem for which the dimension of the function could be increased, usually the search direction becomes orthogonal to the gradient at the point from which search is performed. The disconcerting consequence is that the answer returned by the algorithm is not a solution. And this deterioration of the search direction occurs even for the "perfect" function $x^Tx$. Also it was observed that the negative gradient and the search direction become increasingly orthogonal with each decrease in the size of the step tolerance. Finally it was noted that this deterioration continues for every choice of **n** greater than that at which it first occurs. These observations raise serious doubts as to the applicability of the Nelder-Mead simplex algorithm for all but the smallest values of **n**. Popular conclusion is that the Nelder-Mead simplex algorithm is inefficient for problems with "large" number of variables, say n > 10. The main reason of Nelder-Mead algorithm deterioration, according to Torczon, is the simplex distortion. This algorithm rescales the entire simplex as the last resort. If no improvement can be found by taking any other step, the algorithm takes a "shrink" step. It is possible to show that if the original simplex and its reflection are in a region where the function is convex, then the Nelder-Mead simplex algorithm will not consider the shrink step. As Torczon reported, even when the functions were not convex, only 33 out of some 2.9 million iterations resulted in a shrink step. Thus, if the initial simplex is too large, the Nelder-Mead simplex algorithm only contrasts one vertex at each iteration - and each contraction results in a distortion of the initial simplex. As the dimension of the problem grows, resizing the entire simplex requires more and more iterations, each of which is conducted independently. As a result, the distortion of simplex increases with the dimension of the problem. This distortion in turn means that it is ever more likely that the search directions will deteriorate. And this conjecture is consistent with observations made by both Parkinson and Hutchinson (1972) and Walmsley (1981). An interesting example was reported by Wright (1995): when minimizing $\sum_{i=1}^{n} x_i^2$ with n=32 for one initial configuration, the condition number of the simplex edge matrix never exceeds 150, and all sides of the simplex range between 0.5 and 0.9 in length; nonetheless, the best function value hovers at approximately 0.25 for several thousand iterations. The simplex in this case, although well-behaved, has in effect become misoriented with respect to the contours of **f**.

From the viewpoint of practice, optimization researchers are increasingly interested in the kinds of applications where every function evaluation is precious and costly. Direct search methods constitute one of the few viable alternatives for addressing such problems, especially those in which 'improvement' is sought; hence direct search methods that combine theoretical soundness with both reliability and efficiency are needed. It is still unclear which direct search methods will ultimately emerge as the most effective. Exactly as in large-scale optimization, the 'best' method is likely to be problem dependent. Direct search methods offer opportunities today not only for enlightening new theory, but also for successful solution of important real-world problems. Margaret Wright (1995), who is the author of this analysis, believes that direct search methods deserve further serious attention from the optimization community, and that direct search methods are once again respectable.

### ***Numerical Example***

A single-branch duct system constructed from a ***N*** round duct sections connected in series to a fan is analyzed. The ***N*** number of sections is assumed as 2, 5, 10, and 20. Local resistances are not included for simplicity. Also it is assumed that the flow regime in all sections is quadratic.

The objective function $F$ is the sum of the yearly loan payment for duct construction and the cost of electricity.

$$F = \sum\nolimits_{i=1,N} \left[ 2 * \boldsymbol{p} * R * L * A * I + 0.25 * f * L * \boldsymbol{r} \left( \frac{Q}{p} \right)^2 Q * EC * T / R^5 \right] \qquad (13)$$

where:

*L* - section length = 1.0 m,<br>
*R* - duct section hydraulic radius (optimization parameter),
*A* - Price of the 1 $m^2$ of the surface of the duct = 30 $\$/m^2$,
*I* - yearly loan payment = 10%,
*f* - coefficient of hydraulic resistance = 0.025,
*ñ* - air density = 1.2 $kg/m^3$,
*Q* - flow rate = 0.1 $m^3/sec$,
*T* - work-hours per year = 8760,
*EC* - electricity cost = \$0.04 kW-h.

It is assumed that the sum of total pressure losses at all duct sections is exactly equal to the fan total pressure. The optimization was performed using GenOpt and Newton methods. The comparison is made for the number of function evaluations and the number of iterations. In our case every function evaluation is the simulation task solution. If an airflow system has several hundreds of nodes the simulation task can be time consuming. For the Newton method every iteration needs the solution of two simulation tasks - main task and adjoin task. Adjoin task solution needs approximately the same balk of calculations as the main task. The precision was taken equal 1.0e-4 for all cases.

The results of calculations:

N=2

| GenOpt | Newton method | |
|---|---|---|
| Number of function evaluations | Number of iterations | Number of function evaluations |
| 40 | 7 | 14 |

N=5

| GenOpt | Newton method | |
|---|---|---|
| Number of function evaluations | Number of iterations | Number of function evaluations |
| 126 | 7 | 14 |

N=10

| GenOpt | Newton method | |
|---|---|---|
| Number of function evaluations | Number of iterations | Number of function evaluations |
| 383 | 7 | 14 |

N=20

| GenOpt | Newton method | |
|---|---|---|
| Number of function evaluations | Number of iterations | Number of function evaluations |
| 1048 | 7 | 14 |

The optimum value of the duct radius is equal 0.20591m for all duct sections.

***Conclusion***

GenOpt is a well designed packet for optimization by direct search methods. The first algorithm included into the GenOpt Nelder-Mead method is one of the best direct search methods. It is well balanced between affectivity and robustness. As an example, multidirectional search is more robust but also has dramatically increased number of function evaluations during iteration.

The task for ventilation system optimization, without doubt, is not in the field of the GenOpt application. The ventilation system optimization is the kind of optimization problem involving systems governed by differential equations and other state relations. More precisely, the general nonlinear programming problem

$$\begin{aligned} &\text{minimize } F(a) = f(a,u(a)) \\ &\text{subject to } C_E(a,u(a)) = 0 \qquad\qquad (14) \\ &\qquad\qquad C_I(a,u(a)) >= 0 \end{aligned}$$

with the distinguish feature that **u(a)** is the solution of some set of equations

$$h(a,u(a)) = 0 \qquad\qquad (15)$$

The equation (15) typically describes the physical state of the problem. The most generally effective optimization algorithms for problems such as these are quasi-Newton methods (Dennis 1983), which require derivatives of the objective function F and the constraints.

The adjoint approach (Lions 1971) allows one to compute the derivative F'(a) of F(a) in a very efficient manner. The primary cost in computing F'(a) via the adjoint approach is the calculation of an intermediate quantity ë, called the co-state or adjoint state. Adjoint state is a solution of the adjoint problem, which is a linear problem associated with the governing equation (15). However, the adjoint approach presents certain difficulties. The difficulties are particularly associated with the calculation of the intermediate quantity ë. One problem that arises is determining of the appropriate adjoint approach. It is not always easy to identify the adjoint problem as some manner of conventional initial-value, boundary-value, or initial-boundary-value problem. Objectives in optimization problem for which one cannot identify the appropriate adjoint problem in a straightforward way are sometimes called "inadmissible". The main reason of the inadmissibility - absence of suitable boundary term. Another question associated with the adjoint approach is the convergence of numerical approximation of the cost
at **ë**. Finally, from the point of view of numerical optimization, there is the paramount question of the approximation of the derivative F'(a) and the convergence of such approximations under refinement of discretization of the rank (1-2). On the other hand, the weak form of the adjoint problem always exists. Furthermore, we can solve the weak form of the adjoint problem numerically and can derive convergence estimates for **ë** that depend on the convergence estimates for the solution of the forward problem.

The practical details of the implementation of sensitivity calculations for problem governed by differential equations is a large topic and there is a great deal of disagreement particularly over how the adjoint approach should be implemented (Sei 1995, Williamson 1990). Usually the next two derivations of the weak form of the adjoint problem are used: the first is based on implicit differentiation, while the second is based on the variation of the Lagrangian.

As an example a tangent linear model can be used to analyze the impact of small disturbances. For instance, consider a tangent linear model of the advection of temperatures by horizontal currents. If the temperature at one point is changed, this anomaly is transported downstream and broadened by diffusion. In contrast, the adjoint model can be used to analyze the origin of any anomaly. Constructing the adjoint method from the discrete model equations is usually done by defining a Lagrange Function. The derivatives of the Euler-Lagrange equations with respect to the model variables yield the discrete adjoint equations. This method does not require the construction of adjoing operators. However, extensive and cumbersome coding is necessary. The boundary conditions are handled separately in most applications.

On the other hand, it is possible to develop adjoint code directly from the numerical code of the model (Giering 1998). A numerical model is an algorithm that can be viewed as a composition of differentiable functions, each representing a statement in the numerical code. Note that the order of the evaluation of the individual functions is imposed by the algorithm. Differentiation of the composition can be done by applying the chain rule. The resulting multiple product can be computed in different ways: in forward mode the intermediate derivatives are computed in the same order as the model computes the composition, in reverse mode the intermediate derivatives are computed in reverse order. This method is feasible even for highly sophisticated models with complicated boundary conditions. In this approach, a distinct adjoint model code fragment corresponds to each model code statement. The adjoint code fragments are composed in reverse order compared to the model code. For each kind of statement simple rules can be formulated for constructing adjoint statements. This simplifies considerably the adjoint code construction and debugging.

The results of study show that GenOpt can be useful only for small number of parameters for optimization. For real duct optimization tasks, with number of optimization parameters hundreds and thousands a Newton based algorithms are preferable.

### *10.0 Area of Application and Beneficiaries*

BELLAIR will be developed as a software tool that can be used by an HVAC engineer for optimization and modeling of complex laboratory ventilation and air conditioning system. The program shall be user-friendly, allowing one to execute it with minimum knowledge of PC computers. The limitation exists that the computer program will optimize or simulate only a duct system that is served by a single supply air fan. If a building has many supply systems, multiple computer runs have to be performed: one for each supply system. The number of duct sections and equipment units can be limited only by computer memory.

There are a number of potential beneficiaries of the BELLAIR operation, including:

Investors into the BELLAIR development,
Company developing the BELLAIR program,
Company supporting BELLAIR usage,
Consulting company that designs the laboratory,
Owner of the building,
Company leasing the building,
Building maintenance company.

Each of the beneficiaries has its own goals in the development and usage of BELLAIR.

The investor is interested in profiting by selling a large number of program copies at the highest reasonable price. However, if the investor is a public, industry-supported, or governmental institution, the profit can be limited, provided the effect of usage saves energy, increases the quality of production, or improves worker safety.

The use of a computer program for duct system optimization can result in saving life cycle cost which is the sum of electrical energy cost and the owning cost of a duct system. In many cases duct optimization reduces both types of costs. In other cases it can be economically reasonable to reduce the consumption of electrical energy by an increase in the cross-section of ducts. This saves energy cost for the company which is leasing the building but increases the cost for the building owner that is paying for its construction.

### *11.0 Software Development, Testing, and Implementation Plan*

The main platform for the BELLAIR code will be MS Windows 95/98/NT for IBM PC computers and compatibles. Programming language will be C++ with a Visual C++ compiler, version 5.0 or higher. A DEMO of the program BELLAIR will be installed on the Internet.

The following steps have to be carried out for the development and implementation of the computer code:

**Preprocessor,** consisting of interaction routines between the user and software. The program will be developed to be user-friendly, for an inexperienced user having knowledge in duct design but without knowledge in duct simulation. Fitting selections will be organized using a drag-and-drop technique. If possible, the data will be checked during data entry and error messages should describe the error as well as present recommendation for their correction. The error messages will be divided into two classes, fatal and warning. It will be possible to return to any previous screens. Full capability of inserting, copying, moving, and deleting any part of the data will be implemented using data blocking. The help screens will include examples. Graphic representation and selection will be used for selecting fittings. Reasonable application of the graphical input for topology and length input data has to be investigated. Following are the steps for developing the preprocessor:

(1) Architecture and algorithm development,
(2) Topology representing (numerical and/or graphical),
(3) Selection of graphically presented fittings,
(4) Local fitting data base real time development,
(5) Input data variables and their units,
(6) Screens and menus,
(7) Elements and fittings graphics,
(8) Data checking and diagnostic,
(9) On-line error messages,
(10) Data manipulation routines (browse/insert/copy/move/delete),
(11) Information accessed from the database,
(12) Help index, screens, and messages,
(13) Input data listing.

Following are the steps necessary for the **Data Base** development including fittings, equipment, roughness, specifications, and cost data:

(1) Fittings development and population using the ASHRAE DFDB,
(2) Duct equivalent roughness,
(3) Duct mounted element data,
(4) Duct specification and cost estimation data,
(5) Program messages (data driven information),
(6) Graphic representation of fittings and duct mounted elements using DFDB.

**The Optimization Solver** will consist of all calculation routines. It will include three major parts; hydraulics, control, and minimization and include the following development steps:

(1) Program architecture,
(2) Algorithm,
(3) Local fitting library access,
(4) Element data access,
(5) Fitting tables access and approximation,
(6) Duct hydraulics,
(7) Heat loss and air leakage,
(8) Minimization method calculation,
(9) Specification and cost estimation,
(10) Output data storing.

**The Simulation Solver** will consist of the three following methods: Newton, Tensor, and Homotopy. The selection of the proper method will be performed automatically by the program. Following are the development steps:

(1) Architecture,
(2) Algorithm,
(3) Local fitting library access,
(4) Element data access,
(5) Fitting tables access and approximation,
(6) Duct hydraulics,
(7) Heat loss and air leakage,
(8) Newton method calculation,
(9) Tensor method calculation,
(10) Homotopy method calculation,
(11) Control equipment simulation,
(12) Statistical data selection and recording,
(11) Specification and cost estimation,
(13) Output data storing.

**Postprocessor** will be developed for displaying and printing the results. It will be linked to the Preprocessor. This will allow one to perform interactive studies very efficiently. The development of the Postprocessor includes:

(1) Output screens,
(2) Graphical fan-system characteristics,
(3) Graphical pressure diagrams,
(4) Hard listing of the results.

**Installation** program using WISE will allow the user to install the software on his computer.

**DEMO** program will include the following steps:

(1) Algorithm,
(2) Screens,
(3) Subroutines,
(4) Main program,
(5) HTML/JAVA programming,
(6) Internet server downloading,

**Manual** will include following sections:

(1) Abstract,
(2) License and disclaimer,
(3) Program overview,
(4) Hardware requirements,
(5) Program installation,
(6) Quick start,
(7) Practicing with program,
(8) Run and output,
(9) Examples,
(10) Data verification and validation.

**Testing** [will include] the following steps:

(1) Beta testing,
(2) Alpha testing,
(3) Code correction and modification.

**Program support** is one of the most important parts of the implementation plan. In spite of the fact that development of the software is planned and budgeted, its support should be profit based. It will include the following:

(1) Performing software house functions,
(2) Telephone support,
(3) E-mail/fax written support,
(4) Updating the information.

### *12.0 Software Pricing Strategy*

It is too speculative to predict the profit from any truly new computer program. A general rule shows that the higher the price of the program the smaller the number of customers who will buy it. If the price of the program is low, the number of buyers will be high but the total profit will probably be low due to the relatively high cost for support. Additionally, a low priced program may create a feeling that the quality of the product is inferior and thus will reduce the profit. On the other hand, the total profit for a very high priced program will be low as well due to the limited number of buyers. A curve of price per copy vs. profit, has a maximum point where the first derivative is equal to zero.

Existing commercial duct sizing computer programs are generally priced form $250 to $500 per copy. For example, the program DUCTSIZE developed by Elite Software is priced at $295 per copy. The duct simulation program T-Duct is priced at $495 per copy. If the BELLAIR program were ready, it would be, to the best of our knowledge, the only commercially available duct optimization program in the world. The program will be more comprehensive than any existing duct design software including DUCTSIZE and T-Duct. A proposed price is $695 per copy with unlimited support. Shipping and handling would be extra. Educational copies should be priced at about 40% of a commercial copy or $295. A site network license with up to five (5) users should be priced at $2000 plus $295 per user. A site network license with more than five users should be priced of $2000 plus $195 per user. It is our plan to sell an average of 200 copies per year during the first five years, creating about $600K of total income. This will produce a profit of $400K when support, maintenance, and updates are included.

### *13.0 Funding Sources and Potential Partners*

It is proposed that the necessary funding be obtained from two separate sources. The first source is a grant of $75K from the Energy Innovations Small Grant (EISG) Program administered by the California State University Institute. The second funding source is a $50K supplemental grant from ASHRAE.

The EISG Program is a part of the Public Interest Energy Research (PIER) Program that is managed by the California Energy Commission. The purpose of this Grant Program is to provide benefits to California taxpayers by funding energy research. This Commission provides support to promising new energy concepts.

NETSAL & Associates satisfies the necessary requirement for obtaining the EISG grant since it is a small business located in Fountain Valley, California, and managed by an individual, not an organization. The development of the computer program for life cycle cost optimization of airflow distribution systems proposed by NETSAL is an innovative and original energy concept that addresses a clear market need and will provide benefits for California ratepayers in the Industrial and Commercial areas by reducing energy consumption. A Pre-proposal Abstract (Appendix A) has been sent to EISG. As stated in the PISG Manual, it takes approximately 5 months to complete the proposal evaluation, approval, and agreement execution process. The duration of the development will be 18 month. Then the database will be developed as ASHRAE's part of this project and linked with the preprocessor.

### *14.0 Conclusion*

The concept and implementation plan for the development and implementation of a computer code that can be used by HVAC engineers in the studying, designing, and retrofitting of air distribution systems in research laboratories has been completed. It proposes the development of a computer code that will obtain a minimum life cycle cost by optimally selecting appropriate fans and duct cross-sections. This computer code should be a part of a more general computer program development that combines duct/fan system optimization with TAB (Test and Balancing) simulation and duct/fan system modeling. This more general code will allow HVAC engineers to optimize and analyze supply fan, laboratory spaces, exhaust fans, and ducts as a common generalized system.

There are a number of numerical methods for solving optimization and simulation problems. Duct/fan system optimization requires the use of nonlinear mathematical programming methods such as Reduced gradient, Quadratic search, Dynamic programming, T-Method, etc. Network simulation problem requires a solution for a large system of simultaneous nonlinear algebraic equations. Studied were numerous methods capable of solving nonlinear algebraic equations for both branched and cycled networks simulation. The most promising approaches for cycled systems are the family of Newton Affine Invariant methods, the Tensor methods, and the Homotopy methods. All three approaches are to be included in the simulation computer code and selected automatically during the calculation.

An evaluation was conducted for two software packages, GenOpt and Spark, developed by LANL for nonlinear optimization (GenOpt) and for solving a system of simultaneous nonlinear equations (Spark). This study shows that, in spite of their many positive aspects, neither of these packages is suitable for the optimization program under consideration.

The four main aspects of the proposed computer program are the:

(1) Preprocessor that contains of data input, verification, and printout,
(2) Solver that includes calculation routines,
(3) Postprocessor that performs verification and printout of the results,
(4) Populated database.

The main steps in the development and implementation of the proposed program are:

(1) Create the program architecture,
(2) Perform coding and debugging,

(3) Provide efficient testing,
(4) Develop and populate the data base
(5) Develop Demo and install on the Internet,
(6) Write the user manual,
(7) Provide program installment, update, and support.

The recommended platform for the computer code is MS Windows 95/98/NT for IBM PC computers, C++ programming language, and a Visual C++ compiler.

It is proposed that the necessary funding for program development be obtained from two sources. The first source is a grant from the EISG Program administered by the California State University Institute. The EISG Program is a part of the PIER Program that is managed by the California Energy Commission. The second funding source is a supplemental grant from ASHRAE.

## *15.0 References and Bibliography*

AABC 1965. Principles of Air Distribution, Los Angeles, Associated Air Balancing Council.

AABC 1983. Duct Leakage and Air Balancing. Technical Publication No. 2-83. Washington, DC: Associated Air Balance Council.

Ahmed O., J.W.Mitchell, S.A.Klein 1998. Experimental Validation if Thermal and Pressure Models in a Laboratory Simulator, ASHRAE Transactions 1998, V.104, Pt.2

Allard, F. and Y. Utsumi 1992. Air Flow through Large Openings, Energy and Buildings, Vol. 18, p. 133-146

Altshul, A.D. and P.G. Kiselev 1975. Hydraulics and aerodynamics, 2d ed. Moscow: Stroisdat Publishing House.

Andriyashev, M.M. 1932. Water systems calculation technique, Moscow

AMCA 1973. Fans and Systems, Standard 201-73, Arlington Height, IL

Arioli, M., I.S. Duff, N.I. Gould, and J.K. Reid (1990). Use of the $P^4$ and $P^5$ algorithms for in-core factorization of sparse matrices, SIAM J. Sci. Stat. Comput 11, 913-927.

Arklin, H. and A. Shitzer 1979. Computer Aided Optimal Life-cycle Design of Rectangular Air Supply Duct Systems. ASHRAE Transactions, Vol.85, Part 1, pp. 197-213.

ASHRAE 1997. Fundamentals handbook. Chapter 32, Duct design. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta, GA

ASHRAE 1996. HVAC Systems and Equipment Handbook, Atlanta, GA

ASHRAE 1995. ASHRAE Standard 120P, Laboratory methods of determining flow resistance of air duct and fittings, Atlanta, GA

ASHRAE 1995. HVAC Applications Handbook. Chapter 43. Sound and Vibration Control. Atlanta, GA

ASHRAE 1994. HVAC Duct Fitting Data Base, Research Project 574-RP, Atlanta, GA

ASHRAE 1993. ASHRAE Professional Development Seminars - Air System Design and Retrofit, Atlanta, GA

ASHRAE/SMACNA/TIMA 1985. Investigation of Duct Leakage. ASHRAE Research Project 308.

Axley, J. 1987. Indoor air quality modeling phase II report. U.S. National Bureau of Standards Report NBSIR 87-3661.

Axley, J. 1988. Multi-zone dispersal analysis by element assembly. Submitted to Building and Environment.

Altshul, A.D., and P.G. Kiselev 1975. Hydraulics and aerodynamics, 2d ed. Moscow: Stroisdat Publishing House.

ASHRAE. 1985. ASHRAE Handbook - 1985 Fundamentals, Chapter 33, Duct design. Atlanta: American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc.

ASHRAE. 1989. ASHRAE Handbook - 1989 Fundamentals, Chapter 32, Duct design. Atlanta: American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc.

Baccarat, S.A. 1987. Inter-zone convective heat transfer in buildings: a review. ACME Journal of Solar Engineering, Vol. 109, May.

Baker, P.H. , Sharples, S., and Ward, I.C. 1987. Air flow through cracks. Building and Environment, Vol. 22 No. 4. pp. 293-304.

Baker C.H.T., Paul C.A.H., and Wille D.R. 1995. Issues in the numerical solution of evolutionary delay differential equations, Advanced in Computational Mathematics, 3

Bakhvalov N.S., Knyasev A.V. , and Kobel'kov G.M., 1991. Iterative methods for solving equations with highly varying coefficients. In Roland Glowinski, Yuri A. Kuznetsov, Gerard A. Meurant, Jacques

Periaux, and Olof Widlund, editors, Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, pages 197-205, Philadelphia, PA.

Bellman, R.E. 1957. Dynamic programming. New York: Princeton University Press.

Belsky, J.S. 1991. Design HVAC Systems for Laboratories, Pharmaceutical Engineering, July/August, Vol.11, No.4

Bertschi, R.L. 1969. A Computer Program to Optimize the Life-cycle Cost of a Fan-duct System. Urbana: University of Illinois (thesis).

Berg J. and Lofstrom J., 1976. Interpolation Spaces. An Introduction. Springer-Verlag Berlin Heidelberg.

Besant, Robert W., and Johnson, Alan B. 1995. Reducing Energy Costs Using Run-Around Systems.

ASHRAE Journal, February

Bjorck A., and I.S. Duff. 1980. A direct method for the solution of sparse linear least squares problems. Linear Algebra and Its Applications, 34:43-67

Bjorstad J.H. and J. Mandel 1991. Spectra of sums of ortogonal projections and applications to parallel computing, BIT, 31 (1991), pp. 76-88

Blevins, R.D. 1984. Applied Fluid Dynamics Handbook. New York; Van Norstrand Reinhold.

Bouricha A. 1992. Solving large sparse systems of nonlinear equations and nonlinear least squares problems using tensor methods on sequential and parallel computers. Ph.D. thesis, Computer Science Department, University of Colorado at Boulder

Bouwman, H.B. 1982. Optimal Duct System Design. TNO Research Institute for Environmental Hygiene, Indoor Climate Division, Delft, Netherlands.

Brenan K.E., S.L. Campbell, and L.R. Petzold 1995. Numeric al Solution of Initial-value Problems in Differential-Algebraic Equations, Second Edition, SIAM, 1995.

Brooks, P.J. 1993. New ASHRAE Local Loss Coefficients for HVAC Fittings, ASHRAE Transactions, V.99, Pt.2

Brooks, P.J. 1995. Duct Design Fundamentals, ASHRAE Journal, April

Butakov, S.E. 1949. Aerodynamic of industrial ventilation. Moscow: Profstroisdat Publishing House.

Chow S.N., Mallet-Paret, and Yorke 1978. Finding zeros of maps: homotopy methods that are constructive with probability one, Math.Comp., 32 (1978), pp. 887-899.

Coad, W.J. 1985. Simplified sizing of pipes and ducts, HPAC, July

Coad, W.J. 1980. Energy Concepts of Air Handling Systems, HPAC, August

Colebrook, C.F. 1938-1939. Turbulent flow in pipes with particular reference to the transition region between the smooth and rough pipe laws. Journal of the Institution of Civil Engineers, London, Vol 11.

Coogan J.J. 1996. Effect of Surrounding Spaces on Rooms Pressurized by Different Flow Control ASHRAE Transactions 1996, Vol.102, Pt.1

Cross, H. 1936. Analysis of Flow in Networks of Conduits or Conductors, University of Illinois, Engineering Exp. Station, Bulletin #286, Urbana-Champaign

Data Research 1985. Utility Costs Forecasting, Data Resources Inc., Service Review, McGraw Hill Publishing House.

Davis T.A. and I.S. Duff 1995. A combined unifrontal/multifrontal method for asymmetric sparse matrices, Technical report TR-95-020, Computer and Information Science Department, University of

Florida.

Davis T.A. and I.S. Duff 1993. An asymmetric-pattern multi-frontal method for sparse LU factorization.. Technical report RAL 93-036, Rutherford Appleton Laboratory.

Dennis Jr.,J.E and R.B. Schnabel 1983. Numerical methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall (Englewood Cliffs, New Jersey).

Dennis J.E., J.M.Martinez, X.Zhang 1991. Triangular Decomposition Methods for Solving Reducible Nonlinear Systems of Equations.

Dongarra J.J. 1995. Templates for the Solution of Systems of linear equations. Internet.

Dongarra, J.J., J. Du Croz, J.,I.S. Duff, and S. Hammarling 1990. A set of Level 3 Basic Linear Algebra Subproagrams, ACM Trans. Math.Softw. 16, 1-17.

DFDB 1994. Data Fitting Database. ASHRAE, Atlanta, GA

Duff I.S. and J.K. Reid 1995. MA47, a Fortran code for direct solution of indefinite sparse symmetric linear systems, Technical Report RAL 95-001, Rutherford Appleton Laboratory.

Duff Iain S. 1996. Sparse numerical linear algebra: direct methods and preconditioning'. Report TR-PA-96-22, CERFACS, 42 Ave G Cotriolis, 31057 Toulouse Cedex, France.

Dean R.H., and J.Ratzenberger 1985. Computer Simulation of VAV system Operation, HPAC, September

Dean R.H., F.J.Dean, and J.Ratzenberger 1985. Importance of Duct Design for VAV Systems, HPAC, August

Demuren, A.O. and F.G. Ideriah 1986. Pipe network analysis by partial pivoting method. Journal of Hydraulic Engineering, Vol. 112, pp. 327-333.

Dryja M. and O.B.Wirlund, 1990. Towards a unified theory of domain decomposition algorithms for elliptic problems, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, held in Houston, Texas, March 20-22, 1989.

Dryja M. and O.B.Widlund, 1992. Domain decomposition algorithms with small overlap, Tech. Report 606, Department of Computer Science, Courant Institute, May.

Dongarra Jack J., W.W. David, 1994. Software libraries for linear algebra computations on high performance computers. SIAM.

Durfee, R.L. 1972. Measurement and Analysis of Leakage Rates from Seams and Joints of Air Handling Systems. AISI Project No. 1201-351/SMACNA Project No. 5-71

Energy Innivations Small Grant Program 1999, Grant Application Manual, California Energy Commission, Internet, March, 1999

Evans, R.A. and R.J. Tsal 1996. Basic Tips for Duct Design, ASHRAE Journal, July, 1996

Eisenstat, S.C. and J.W.Liu 1992. Exploiting structural symmetry in asymmetric sparse symbolic factorization, SIAM J. Matrix Analysis and Applications 13, 203-211.

EPA 1995. Electric Sales and Revenue 1994, Energy Information Administration, Washington, DC

EPA 1985. Utility Cost Forecasting, Energy Information Administration, Washington

Farajian, T., G. Grewal, R.J.Tsal 1992. Post-accident Air Leakage Analysis in a Nuclear Facility via T-Method Air flow Simulation. 22nd DOE/NDC Nuclear Air Cleaning and Treatment Conference, Denver

Farguhar, H.E. 1973. System Effect Values for Fans. ASHRAE Symposium Bulletin, Louisville, Kentucky

Fox, R.L. 1971. Optimization Methods for Engineering Design, Addison-Wesley Publishing Company, Reading, MA

Frommer A. and G.Mayer 1989. Convergence of Relaxed Parallel Multisplitting Methods, Linear Algebra Application. 119, pp. 141-152

Furi M. and A.Vignolli 1994. On the regularization on nonlinear ill-possed problem in Banach spaces. - J. Optimization Theory and Application, 4.3.

George A. 1973. Nested dissection of a regular finite element mesh, SIAM J. Numerical Analysis 10, 345-363.

George, A. and L.W.H. Liu 1989. The evolution of the minimum degree ordering algorithm, SIAM Review 31(1), 1-19.

Giering R. , T.Kaminski 1998. Recipes for Adjoint Code Construction. Mux-Planck-Institut fur Meteorologie.

Gilbert, J.R. and T.Peierls 1988. Sparse partial pivoting in time proportional to arithmetic operations, SIAM J. Scientific and Statistical Computing 9, 862-874.

Golub G.H. and C.F.Van Loan 1989. Matrix computations, John Hopkins University Press, second ed.

Graham, J.B. 1996. Duct System Pressure Gradient Diagrams and the Beer Cooler Problem, HPAC, August

Gregory, W.S., K.H. Duerre, and G.A. Bennett 1975. Ventilation system analysis during tornado conditions. Los Alamos Scientific Laboratory, Progress Report LA-6293-PR, July-December.

Gill P.E, Murray W., and Wright M.H. 1983. Practical Optimization. Academic Press.

Hackbusch W. 1985. Multigrid Methods and Applications, vol. 4 of Computational Mathematics, Springer Verlag, Berlin.

Hansen, P.C., T.Ostromsky, and Z.Zlatev 1994. Two enhancements in a partitioned sparse solver, in J.J.Dongarra and J.Wasniewski, eds, Paralle l Scientific Computing. Proceedings of the PARA94 Conference, Copenhagen 1994, Lecture Notes in Mathematics 879, Springer, Berlin, pp. 296-303.

Haves P., and E. F. Sowell, 1999. The Application of Problem Reduction Techniques Based on Graph Theory, to the Simulation of Non-Linear Continuous Systems. LNBL, San Francisco.

Haves P., L.K.Norford, and M. DeSimone, 1998. A Standard Simulation Test Bed for the Evaluation of Control Algorithms and Strategies, ASHRAE Transactions, Vol.104, Pt.1

Hohmann A., 1993. Inexact Gauss Newton Methods for Parameter Dependent Nonlinear Problems. Technical report TR-93-13 (December 1993).

Hitchings, D.T. 1994. Laboratory Space Pressurization Control Systems, ASHRAE Journal, February

Horowitz, E. and S. Sahni 1976. Fundamentals of Data Structures. New York: Computer Science Press.

Houghton D.J., R. C. Bishop, A. B. Lovins, and B. L. Stickney, 1992. Space Cooling and Air Handling, E-Source Inc.report, Boulder, Colorado, August

Huang C.M. and D.P.O'Leary 1994. A Krylov Multisplitting Algorithm for Solving Linear Systems of Equations, Linear Algebra and Its Applications.

Idelchik, I.E. 1996. Handbook of Hydraulic Resistance, 3rd edition, Begell Publishing House, Inc.

Isaacs, L.T., and K.G. Mills 1980. Linear theory methods for pipe network analysis. Journal of the Hydraulics Division, Proceedings, ASCE, July, pp. 1191-1201

Jeppson, R.W. 1976. Analysis of flow in pipe network. Ann Arbor, MI: Ann Arbor Science.

Jacoby S.L.S.Kowalik J.S, and.Pizzo J.T. (1972). Iterative Methods for Nonlinear Optimization Problems. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Kamenev, P.N. 1938. Flow dynamics of industrial ventilation. Moscow: Stroisdat Publishing House.

Kiczales, G. 1996. Beyond the Black Box: Open Implementation. IEEE Software, 13(1): p. 8-11.

Klir George J. 1990. Architecture of systems problem solving. Plenum press. New York and London.

Klote, J.H. 1981. A Computer Program for Analysis of Smoke Control Systems. National Bureau of Standards, NBSIR 80-2157, Maryland

Konstantinov, U.M. 1981. Hydraulics. Kiev, USSR: Vischa Scola Publishing House.

Kovaric, M. 1971. Automatic Design of Optimal Duct Systems. Use of computers for environmental engineering related to buildings. Washington: National Bureau of Standards, Building Science Series 39, (October), pp.385-391.

Lions J.L.(1971) Optimal Control of Systems Governed by Partial Differential Equations, Springer-Verlag,.

Lam, C.F. and M.L.Walla, 1972. Computer Analysis of Water Distribution Systems: Part I - Formulation of Equations, ASCE, Vol. 98.

Larson, E.D. and L.J. Nilsson 1991. Electricity Use in Pumping and Air Handling Systems, ASHRAE Transactions 97(2)

Leah, R.L., Pedersen, C.O, and Liebman, J.S. 1987. Optimization Using Quadratic Search - A Case Study of a Chilled Water System, ASHRAE Transactions, Vol.93, Part 2.

Liddament, M., and Thompson, C. 1982. Mathematical models of air infiltration - a brief review and bibliography. Technical Note AIC 9, The Air Infiltration Centre, Bracknell, England.

Lobaev, B.N. 1959. Duct Calculation. Gosstroy Publishing House, Kiev, USSR. Available from the Library of Congress, Washington D.C.

Lorenzetti D.M. 1999. A Direct Error Measure for Affine Models of Nonlinear Algebraic Systems. National Institute of Standards and Technology, Division 863, Gaitherburg, Maryland. Draft paper.

Lorenzetti D.M. 1999. Regulating Newton-Raphson's Method by Unconstrained Minimization of a Weighted Sum of Squared Residuals. National Institute of Standards and Technology, Division 863, Gaitherburg, Maryland. Draft paper.

Markowitz, H.M. 1957. The elimination form of the inverse and its applications to linear programming, Management Science 3, 255-269.

Martin D.W. and G.Peters 1963. The Application of Newton's Method to Network Analysis by Digital Computer , Journal of Institute of Water Engineers, Vol.17

McCormick S.F. 1992. Multilevel Projection Methods for Partial Differential Equations, vol. 6 of Frontiers in Applied Mathematics, SIAM, Philadelphia.

McDiarmid, Michael D. 1990. Variable-Volume Fume Hoods: User Experience in a Chemistry Research Laboratory. ASHRAE Transactions, Vol. 96, Part 2.

Merenkov,A.P, 1992. Mathematical modeling and optimization of heating/water/oil and gas supply systems, Science Publishing House, Siberian branch, Novosibirsk, Russia.

Meyer, M.L. 1973. A New Concept: The Fan System Effect Factor, ASHRAE Symposium Bulletin, Louisville, KY

Moody, L.F. 1944. Friction factor for pipe flow. ASME Transactions, Vol.66, p.671.

Murtagh, B.A. 1972. An Approach to the Design of Networks. Chemical Engineering Science, Permagon Press, Vol.27, pp.1131-1143

Murphy G.C. 1976. Practical Aspects of Ductwork Design, ASHRAE Journal, October

Nelder J.A. and Mead R. 1965. A simplex method for function minimization. The computer Journal, 7(4):308-313, January 1965.

Neuman, V.A., and H.M.Guven 1988. Laboratory Building HVAC Systems Optimization. ASHRAE Transactions 1988, Vol. 94, Part 2.

Neuman, V.A. and W.H. Rousseau 1986. VAV for Laboratory Hoods - Design and Cost. ASHRAE Transections, Vol. 92, Part 1A.

Nocedal J. and S.Wright 1998. Numerical Optimization. Internet

Novak U., K.Weimann 1991. A Family of Newton Codes for Systems of Highly Nonlinear Equations. Technical Report TR-91-10 (December 1991).

Nillson, L.J. 1993. Energy Systems in Transition, Department of Environmental and Energy Systems Studies, Lund, Sweden

Numerical methods for Nonlinear Algebraic Equations. Edited by P.Rabinowitz. Gordon and Breach science publishers 1970.

Ortega J.M. and W.C.Rheinboldt 1970. Iterative solution of nonlinear equations in several variables. New York: Academic Press.

Ostrowski, A. 1966. Solution of equations and systems of equations, 2nd ed. New York: Academic Press.
Peters G., and J.H. Wilkinson. The least squares problem and pseudo-inverses. Computer J., 13:309-316,1970.

Parkinson L.M., and Hutchinson D. 1972. An investigation into the efficiency of variants on the simplex method. In F.A. Lotsma, editor, Numerical Methods for Non-linear Optimization, pages 115-135. Academic Press, London and New York, 1972.

Petzold L.R., K.E. Brenan, S.L. Campbell. 1989. The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations, Elsevier Science Publishing Co.

Rabinowitcz 1970. Numerical methods for Nonlinear Algebraic Equations, Gordon and Breach science publishers.

Reid, J.K. 1987. Sparse matrices, in A. Iserles and M.J.D. Powell, eds, The State of the Art in Numerical Analysis, Oxford University Press, Oxford, pp. 59-85.

Rheinboldt W.C. 1982. On the computation of critical boundaries on equilibrium surfaces, SIAM J. Numer. Anal., 18 (1982), pp.653-669.

Rheinboldt W.C. and J.V. Burkardt 1983. Algorithm 596: A program for a locally parameterized continuation process, ACM Trans. Math. Software, No.9, pp. 236-241.

Rozell, J.M. 1974. Duct Turning Vanes in 90° Elbows, ASHRAE Transactions 80(2):53

Said, M.N.A. 1988. A review of smoke control models. ASHRAE Journal, Vol. 30, No 4, April.

Scott K.B., 1986. Don't Ignore Duct Design for Optimized HVAC Systems, Specifying Engineer, No. 62, January

Sei A. and W.W.Symes 1995. A note onn consistency and adjointness for numerical schemes, Tech.

Report TR95-04, Department of Computational and Applied Mathematics, Rice University, January

Shao J.P. 1993. Domain Decomposition Algorithms, Ph.D. Thesis, University of California, Los Angeles

Shifrinson, B.L. 1937. New Method for District Water System Optimization, Heat and Power, No.2, Moscow, (Feb.), pp. 4-9. Available from the Library of Congress, Washington D.C.

SMACNA 1995. HVAC Duct Construction Standards - Metal and Flexible, 2nd Edition, Sheet Metal and Air Conditioning Contractor's Association, Chantilly, VA

SMACNA 1990. HVAC Systems - Duct Design, third edition

SMACNA 1985. HVAC Air Duct Leakage Test Manual

SMACNA 1975. Industry Practice for Industrial Duct Construction

SNIP 1976. II-33-75, Gosstroy, Stroysdat, Part II, Chapter 33, Moscow, USSR

SPARK - Simulation Problem Analysis and Research Kernel (SPARK), Internet 1999

Stepanov A. and M.Lee 1995. The Standard Template Library . Hewlett-Packard Laboratoies.

Stoecker, W.F., R.C. Winn, and C.O. Pedersen 1971. Optimization of an Air-supply Duct System. Use of computers for environmental engineering related to buildings. National Bureau of Standards, Building Science Series 39, October

Stoecker, W. F., L. P. Rinck, and J. P. Wegrzyn 1974. Simulating flow rates, pressures, and temperatures in central chilled water systems. Second Symposium of the Use of Computers for Environmental Engineering Related to Buildings, Paris, June 13-15.

Swami, M.V., and S.Chandra. 1988. Correlations for pressure distribution on buildings and calculation of natural-ventilation air flow. ASHRAE Transactions, Vol. 94, Part 1.

Swim, W.B. 1984. Analysis of Duct Leakage ETL Data from ASHRAE RP 308

Schnabel, R.B. and P.Frank 1984. Tensor methods for nonlinear equations, SIAM J. Numer. Anal., 21, pp.815-843

Tihonov A.N., V.Y.Arsenin 1986. The methods for ill-posed problems solution, Moscow, Nauka

T-DUCT Computer Program Manual 1994. Netsal & Associates, California

Torczon V. 1989. Multi-Directional Search: A Direct Seacrh Algorithm for Parallel Machines. Ph.D thesis, Rice University, Houston, Texas, USA.

Tsal R.J., J. Bell 1998. Laboratory Air Distribution. Laboratory Air Distribution Design Guide, CIEE and Lawrence Berkeley National Laboratory, San-Francisco, 1998

Tsal, R.J., H.F. Behls, and R. Mangel 1988. T-Method Duct Design, Part I: Optimization Theory, ASHRAE Transactions, Vol. 92, Part 1A

Tsal, R.J., H.F. Behls, and R. Mangel 1988. T-Method Duct Design, Part II: Calculation Procedure and Economic Analysis, ASHRAE Transactions, Vol. 92, Part 1A

Tsal, R.J., H.F. Behls, R. Mangel 1990. Duct Design Part, III: T-Method Duct Simulation. ASHRAE Transactions, Vol. 96, Part 2

Tsal, R.J., H.F. Behls, L.P.Varvak, 1998. T-Method Duct design. Part IV: Duct Leakage Theory, ASHRAE Transactions, Vol. 104, Part 2.

Tsal, R.J., H.F. Behls, L.P.Varvak 1998. T-Method Duct Design, Part V: Duct Leakage Calculation Technique and Economics. ASHRAE Transactions, Vol. 104, Part 2.

Tsal, R.J. and L.P. Varvak 1992. Duct Design Using the T-Method with Duct Leakage Incorporated. ASHRAE Research Project 641-RP

Tsal, R.J. and H.F. Behls 1990. Using the T-Method for Duct System Design, ASHRAE Journal, March

Tsal, R.J. and H.F. Behls 1989. Ducting In Technology Menu for Efficient End Use of Energy, University of Lund, Gerbagatan Lund, Sweden

Tsal, R.J. and P.W. Othmer 1989. New Method for Duct Design, CLIMA 2000. The Second International Congress on Heating, Refrigerating and Air Conditioning, Saraevo, Yugoslavia

Tsal, R.J. 1988. Calculation Techniques for Optimum Duct Design and Flow Simulation, ASHRAE Research Project 516-RP and Fluor Research Project 321563, Irvine, CA

Tsal, R.J., and H.F. Behls 1988. Fallacy of the Static Regain Duct Design Method, ASHRAE Technical Data Bulletin, ASHRAE, Atlanta, GA

Tsal, R.J. and M.S. Adler 1987. Evaluation of Numerical Methods for Ductwork and Pipeline Optimization, ASHRAE Transactions, Vol.93, Part 1

Tsal, R.J. and H.F. Behls 1986. Evaluation of Duct Design Methods. ASHRAE Transactions, Vol. 92, Part 1A

Tsal, R.J. and E.I. Chechik 1968. Use of Computers in HVAC Systems. Kiev: Budivelnick Publishing House. Available from the Library of Congress, Washington D.C. ,service number TD153.T77

Tsal, R.J. and N.Z. Shor 1967. The Use of the Steepest Descent Method for Calculating Flow Distribution in a Duct System. Economical Cybernetics and Operation Research, No.2. Institute of Cybernetics Publication, Kiev, USSR

Tsao Tsu-Chin 1992. Variable Air Volume Heating, Ventilation, and Air Condition (HVAC) Control System Model, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana-Champaign, Technical report UILU-ENG 92-4032, Illinois

Yserentant H. 1990. Two preconditioners based on the multi-level splitting of finite element spaces, Numer. Math., 58, pp 163-184.

Young D. 1971. Iterative Solution of Large Linear Systems. Academic Press, New York

Varga R.S. 1962. Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs, N.J.

Vicente L.N. 1996. Trust-Region Interior Point Algorithms for a Class of Nonlinear Programming Problems, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas

Walton, G.N. 1982. Air Flow and Multi-room Thermal Analysis, ASHRAE Transactions, Vol. 88, Part 2.

Walton, G.N. 1984. A Computer Algorithm for Predicting Infiltration and Interroom Air Flows. ASHRAE Transactions, Vol. 90, Part 1.

Walton, G.N. 1989. AIRNET - a Computer Program for Building Air flow Network Modeling, NISTIR 89-4072, National Institute of Standard and Technology, Maryland

Watson L. T. 1986. Numerical Linear Algebra Aspects of Globally Convergent Homotopy Methods. SIAM Review Vol.28, No.4, December

Watson L. T. , S.C. Billups, and A.P. Morgan 1985. HOMPACK: A suite of codes for globally convergent homotopy algorithms, Tech. Rep. 85-34, Dept. of Industrial and Operations Eng., Univ. of Michigan, Ann Arbor, MI, 1985.

Wendes 1986. Sheet Metal Estimating, Wendes Engineering and Constructing Services, Inc.

Williams, G.J. 1995. Air Systems Basics, Heating/Piping/Air Conditioning, May

Walmsley D.A. 1981. The simplex method for minimization of a general function. Supplementary Report 686, Assesment Division, Transport Systems Department, Transport and Road Research Laboratory, Crowthorne, Berkshire, 1981.

Williamson K.A. 1990. Parameter Identification in Systems of Ordinary Differential Equations. PhD thesis, Rice University, Houston, Texas, 1990.

Woods D.J. 1985. An Interactive Approach for Solving Multi-objective Optimization Problems. PhD thesis, Rice University, Houston, Texas, USA.

Wood, D.J., and A.G.Rayes 1981. Reliability of algorithms for pipe network analysis. Journal of the Hydraulics Division, Vol. 107, pp. 1145-1161.

Wright Margaret H. (1995) Direct Search Methods: Once Scorned, Now Respectable, in D.F. Griffits and G.A. Watson (eds), Numerical Analysis 1995 (Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis), 191 - 208, Addison Wesley Longman, Harlow, United Kingdom.

## *16.0 Appendix A*

### Computer Program for Airflow Distribution Systems Optimization
### Proposal for Small Grant by EISG Program
### Pre-proposal Abstract

NETSAL @ Associates,
16182 Mount Lowe Circle, Fountain Valley, CA 92708
Tel.: 714-531-2960, E-mail: netsal@apc.net, Web: www.apc.net/netsal

Heating, ventilating and air conditioning (HVAC) systems are the major energy consumers in most industrial and commercial buildings. The HVAC system cost can exceed 30% of the first cost of a building and 40% of the building's operating energy expense. Proper system design is one of the ways of reducing energy use and life-cycle costs as well as enhancing the performance of HVAC systems. According to American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) Technical Data Bulletin ("T-Method Duct Design," Ottawa, Ontario, June 1988), the savings provided by using the
T-Method for the life-cycle cost optimization of the duct system presented as an ASHRAE Handbook example when located in a residential building in New York City is 53.4%.

High quality, multi-fan system optimization can be achieved only by using a comprehensive computer program. However, the existing commercial duct design computer codes are still based on arbitrary engineering procedures, such as equal friction or static regain, and do not accomplish life-cycle cost and energy optimization. As yet there is no practical duct design program that addresses economic system optimization. This project is targeted to the development and implementation of such a computer program. The computer program will be developed to be used by HVAC engineers in the design of new and retrofit air distribution systems in industrial and commercial buildings.

The goal of this project is to develop a computerized numerical model, which is a computer program capable of performing an economic optimization of a complex HVAC system that includes both supply and exhaust systems. This computer program will select fans and duct cross-sections and provide for airflow resistances caused by duct mounted equipment. The program will use the now available powerful numerical optimization methods. This approach will save electrical energy and reduce first and operating costs for the facility. Constant air volume and variable air volume central supply systems with multi-fan, individual or manifolded exhaust systems will all be considered.

A candidate tool for solving the nonlinear minimization problems necessary to optimize airflow distribution systems is the T-Method of duct design developed by NETSAL for ASHRAE in 1988 and the GenOpt software developed by Lawrence Berkeley National Laboratory this year. It is planned that the proposed computer program be distributed through the Internet to serve companies that are involved in the design, testing, and balancing of HVAC systems.

There are four major elements to the project: (1) the **preprocessor** that consists of interactive data input and verification; (2) the equation **solver** that includes calculation routines; (3) the **postprocessor** that performs verification and printout of the results; and (4) the populated **database**.

The main platform for the computer code is MS Windows 95/98/NT for PC computers. Programming language will be C++ and/or Visual Basic. A Demo of the program will be installed on the Internet.

Development of the computer program will fall below the $75,000 cost budget limit of EISG and require 16-18 months to complete.  ASHRAE research is funding a continuation of this project to update the ASHRAE Duct Fitting Database to be more user-friendly and more complete. The ASHRAE research program will be needed to fully populate the database for this proposed computer program.

Robert J. Tsal, Ph.D.
Vice President